十速

TM57PE40

DATA SHEET

Rev V1.9

tenx reserves the right to change or discontinue the manual and online documentation to this product herein to improve reliability, function or design without further notice. **Tenx** does not assume any liability arising out of the application or use of any product or circuit described herein; neither does it convey any license under its patent rights nor the rights of others. **Tenx** products are not designed, intended, or authorized for use in life support appliances, devices, or systems. If Buyer purchases or uses tenx products for any such unintended or unauthorized application, Buyer shall indemnify and hold tenx and its officers, employees, subsidiaries, affiliates and distributors harmless against all claims, cost, damages, and expenses, and reasonable attorney fees arising out of, directly or indirectly, any claim of personal injury or death associated with such unintended or unauthorized use even if such claim alleges that tenx was negligent regarding the design or manufacture of the part.

AMENDMENT HISTORY

Version	Date	Description
V1.0	May, 2011	New release
V1.1	Oct, 2011	Modify the package type data.
V1.2	Dec, 2011	Add Ordering Information table in the Packaging Information section.
V1.3	Jan, 2012	1. Add the Electrical Characteristics specs in the Features section. 2. Add description in Reset section.
V1.4	Mar, 2012	1. Modify description in Touch Key section. 2. Modify the figure in Touch Key using Timer0 and Timer1 timer.
V1.5	Mar, 2013	1. Modify TM57PE40 Block Diagram. 2. Modify Pin Description Section. 3. Modify Status Register description. 4. Modify PA3/PA4 IO setting notes in dual clock mode. 5. Modify WDT/WKT Block Diagram. 6. Modify LVR Circuit Characteristics data.
V1.6	Jun, 2013	1. Add supported EV board on ICE. 2. Add Pin Summary. 3. Modify Ordering Information.
V1.7	Aug, 2013	Modify Ordering Information
V1.8	Sep, 2015	Modify Operating Voltage range
V1.9	Mar, 2018	Modify Package type

CONTENTS

FEATURES

1. **ROM: 4K x 14 bits OTP or 2K x 14 bits TTP™ (Two Time Programmable ROM)**
2. **RAM: 176 x 8 bits**
3. **STACK: 6 Levels**
4. **Oscillation Sources**
   - Fast-clock
     - FXT (Fast Crystal): 1M~24 MHz
     - FIRC (Fast Internal RC): 2/4/8/16 MHz
     - XRC (External R, External C): 10K~3 MHz
   - Slow-clock
     - SXT (Slow Crystal): 32768 Hz
     - XRC (External R, External C): 10K~3 MHz
     - SIRC (Slow Internal RC): 168K/40K/9.8K/2.6 KHz @5V; 128K/30.3K/7.6K/2K @3V
     - TKRC (Touch Key Clock): 128K/64K/16K/4 KHz @5V, un-touched; 80K/40K/10K/2.5K @3V, un-touched
5. **Dual System Clock**
   - FIRC + SIRC
   - FIRC + SXT
   - FIRC + XRC
   - FIRC + TKRC
   - FXT + SIRC
   - FXT + TKRC
   - XRC + SIRC
   - XRC + TKRC
6. **Power Saving Operation Mode**
   - FAST Mode: Slow-clock can be disabled or enabled
   - SLOW Mode: Fast-clock stops, CPU running
   - IDLE Mode: Slow-clock running, CPU stops, Timer2 is running
   - STOP Mode: All Clocks stop, Wake-up Timer is disabled or enabled
7. **Operation Voltage and Speed: VDD=1.6V @4 MHz**

8. 3 Independent Timers

- Timer0
  - 8-bit timer divided by 1~256 pre-scaler option, Counter / Interrupt / Stop function
  - Capture – high duty or low duty (pulse width measurement)
  - Overflow and Toggle out
- Timer1
  - 16-bit timer with two pre-scalers, Counter / Interrupt / Stop / Clear&Hold / Set / Reload function
  - Capture – period time
  - Overflow and Toggle out
- Timer2
  - 15-bit timer with 4 interrupt interval time options
  - IDLE mode wake-up timer or used as one simple 15-bit timer base
  - Clock source: SXT / XRC / SIRC / TKRC

9. Interrupt

- Three External Interrupt pins
  - 2 pins are falling edge wake-up triggered
  - 1 pin is rising or falling edge wake-up triggered
- Timer0 / Timer1 / Timer2 / WKT (wake-up) Interrupts
- Comparator output change interrupt

10. PB[7:0] individual pin low level wake up

11. Wake-up (WKT) Timer

- Clocked by built-in RC oscillator with 4 adjustable Interrupt times
  0.9 ms/1.8 ms/30 ms/120 ms @5V, 1 ms/2 ms/32 ms/128 ms @3V

12. Watchdog Timer

- Clocked by built-in RC oscillator with 4 adjustable Reset Time
  100 ms/200 ms/800 ms/1600 ms @5V, 130 ms/280 ms/1100 ms/2200 ms @3V
- Watchdog timer can be disabled/enabled in STOP mode (WDTSLPSTP, R0Eh.5)

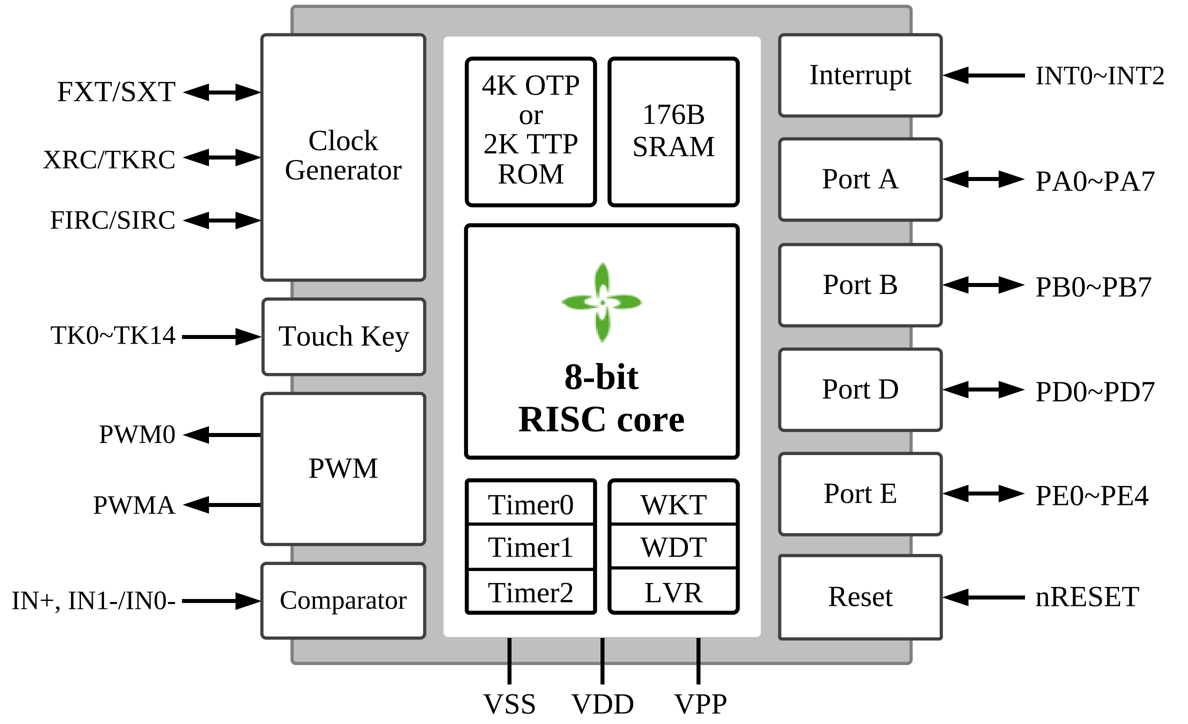
13. 2 Independent PWMs

- PWM0:
  - 8-bit with 1~8 pre-scalers, period-adjustable / duty-adjustable / Clear&Hold / Non-inverting or inverting output.
- PWMA:
  - 8+2 bits, duty-adjustable controlled PWM

14. **One analog voltage comparator**
15. **15-channel Touch Key, supports one key wakeup for low power consumption**
16. **Reset Sources**
    - Power On Reset / Watchdog Reset / Low Voltage Reset / External Pin Reset
17. **Low Voltage Reset Option: LVR1.5V, LVR1.5V disable in SLEEP, LVR2.3V, LVR3.2V**
18. **Operation Voltage: Low Voltage Reset level to 5.5V**
    - fosc = 4 MHz, 1.7V ~ 5.5V
    - fosc = 8 MHz, 1.8V ~ 5.8V
    - fosc = 12 MHz, 2.1V ~ 5.5V
    - fosc = 16 MHz, 3.1V ~ 5.5V
    - fosc = 24 MHz, 4.0V ~ 5.5V
19. **Operating Temperature Range: -40°C to +85°C**
20. **Instruction set: 36 Instructions**
21. **Instruction Execution Time**
    - 2 oscillation clocks per instruction except branch
22. **I/O ports: Maximum 29 programmable I/O pins**
    - Pseudo-Open-Drain Output
    - Open-Drain Output
    - CMOS Push-Pull Output
    - Schmitt Trigger Input with pull-up resistor option
23. **Package Types:**
    - 24-pin SOP (300 mil), SSOP (209 mil)
    - 28-pin DIP (300 mil), SOP (300 mil), SSOP (209mil)
    - 32-pin SOP (300 mil), QFN (4x4x0.75-0.4mm)
24. **Supported EV board on ICE**

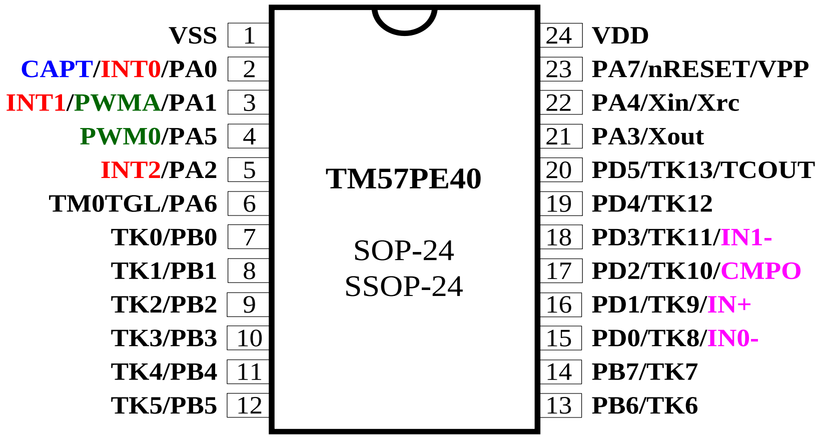
    EV board: EV2787

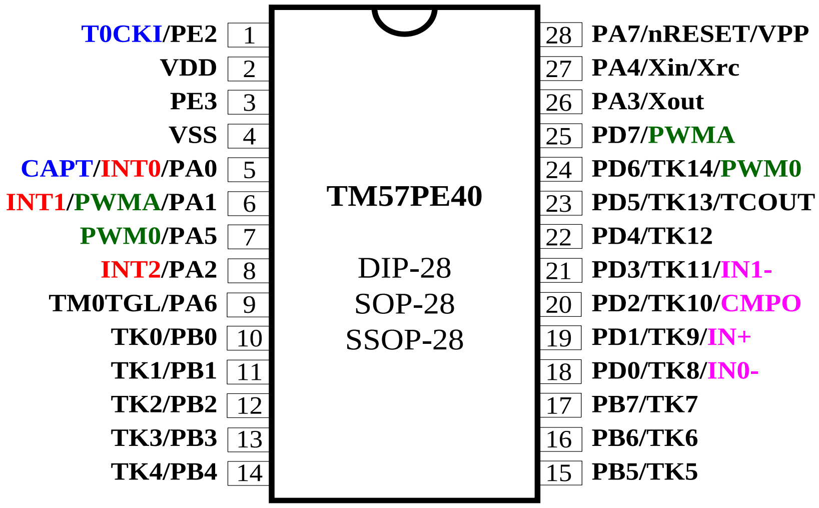
BLOCK DIAGRAM

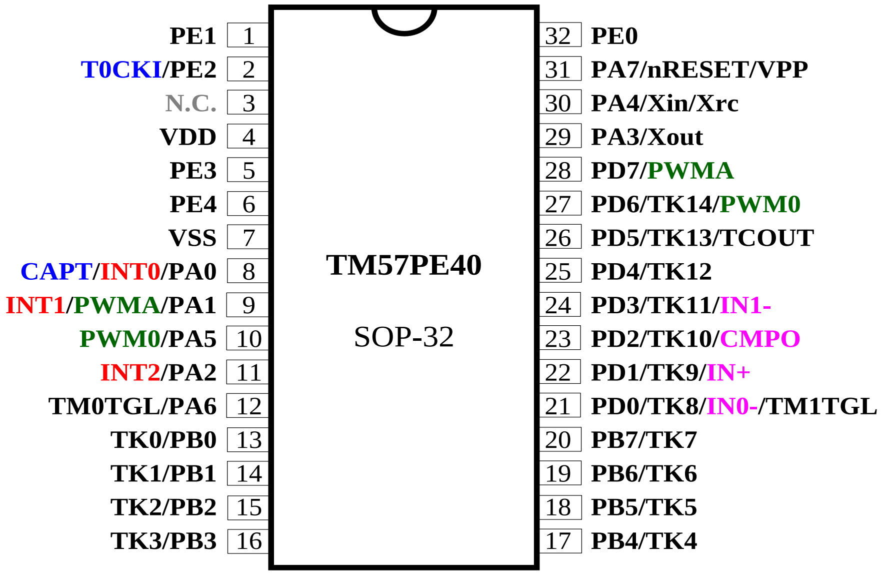


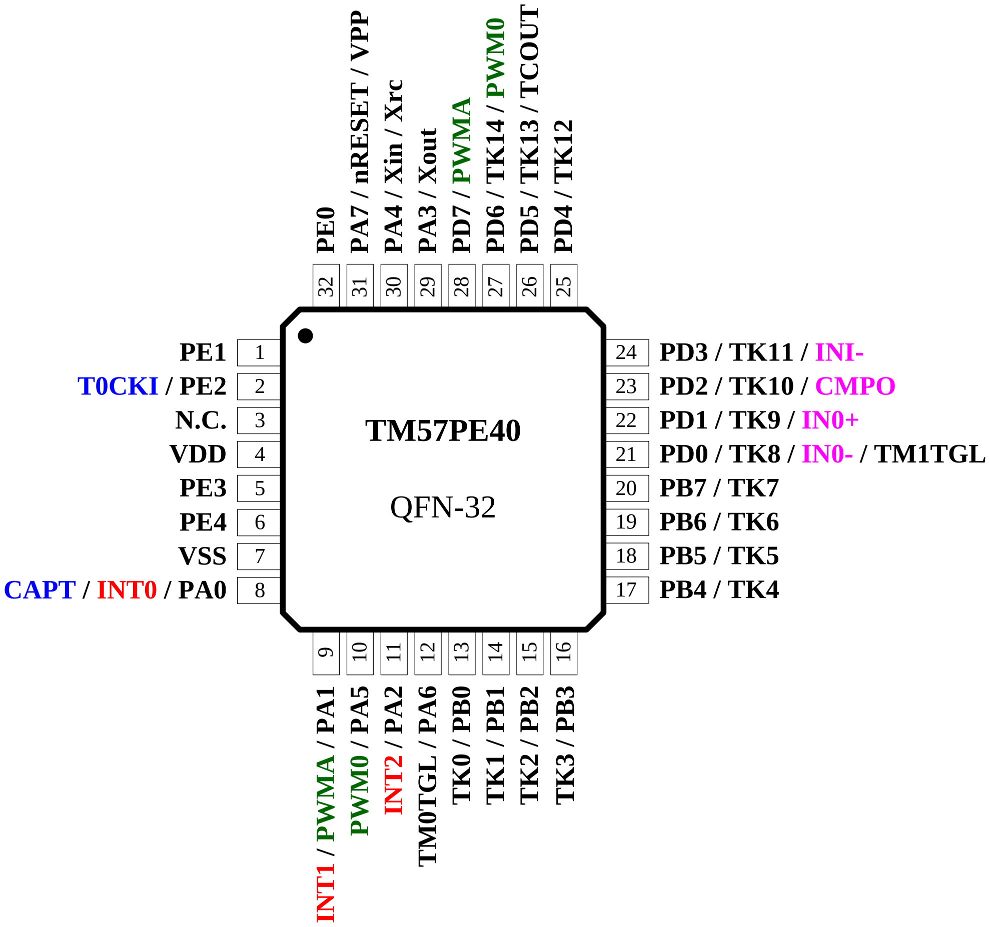
TM57PE40 Block Diagram

PIN ASSIGNMENT








TM57PE40
QFN-32
1 PE1
2 T0CKI / PE2
3 N.C.
4 VDD
5 PE3
6 PE4
7 VSS
8 CAPT / INT0 / PA0
9 INT1 / PWMA / PA1
10 PWM0 / PA5
11 INT2 / PA2
12 TM0TGL / PA6
13 TK0 / PB0
14 TK1 / PB1
15 TK2 / PB2
16 TK3 / PB3
17 PB4 / TK4
18 PB5 / TK5
19 PB6 / TK6
20 PB7 / TK7
21 PD0 / TK8 / IN0- / TM1TGL
22 PD1 / TK9 / IN0+
23 PD2 / TK10 / CMPO
24 PD3 / TK11 / INI-
25 PD4 / TK12
26 PD5 / TK13 / TCOUT
27 PD6 / TK14 / PWM0
28 PD7 / PWMA
29 PA3 / Xout
30 PA4 / Xin / Xrc
31 PA7 / nRESET / VPP
32 PE0

PIN DESCRIPTION

Name	In/Out	Pin Description
PA0–PA2	I/O	Bit-programmable I/O port for Schmitt-trigger input, CMOS push-pull output or "**pseudo-open-drain**" output. Pull-up resistors are assignable by software.
PA3–PA6	I/O	Bit-programmable I/O port for Schmitt-trigger input, CMOS push-pull output or "**open-drain**" output. Pull-up resistors are assignable by software.
PA7	I	Schmitt-trigger input with pull-high
PB0–PB7 PD0–PD7 PE0–PE4	I/O	Bit-programmable I/O port for Schmitt-trigger input, CMOS push-pull output or "**open-drain**" output. Pull-up resistors are assignable by software.
nRESET	I	External active low reset
Xin, Xout	–	Crystal/Resonator oscillator connection for system clock.
Xrc	–	External RC oscillator connection for system clock
VDD, VSS	P	Power Voltage input pin and ground
VPP	I	PROM programming high voltage input
INT0–INT2	I	External interrupt input
PWM0/PWMA	O	PWM output
TCOUT	O	Instruction cycle clock divided by N output. Where N is 1,2,4,8. The instruction clock frequency is system clock frequency divided by two (Fcpuclk/2).
T0CKI	I	Timer0's input in counter mode
CAPT	I	Timer0/Timer1 Capture input
TK14-TK0	I	Touch Key Input
IN+	I	Comparator positive input
IN1-/IN0-	I	Comparator negative input, selected by CMPINNS bit
TM0TGL	O	Timer0 overflow toggle output
TM1TGL	O	Timer1 overflow toggle output
CMPO	O	Comparator output

PIN SUMMARY

Pin Number			Pin Name	Type	GPIO					Function After Reset	Alternate Function			
					Input		Output							
32-SOP/DIP	28-SOP/DIP	24-SOP/DIP			Weak Pull-up	Ext. Interrupt	O.D	P.O.D	P.P		PWM	Touch Key	ADC	MISC
1	-	-	PE1	I/O	○		○		○					
2	1	-	T0CKI/PE2	I/O	○		○		○					T0CKI
3	-	-	N.C.	-										
4	2	24	VDD	P										
5	3	-	PE3	I/O	○		○		○					
6	-	-	PE4	I/O	○		○		○					
7	4	1	VSS	P										
8	5	2	CAPT/INT0/PA0	I/O	○	○		○	○					CAPT
9	6	3	INT1/PWMA/PA1	I/O	○	○		○	○		○			
10	7	4	PWM0/PA5	I/O	○		○		○		○			
11	8	5	INT2/PA2	I/O	○	○		○	○					
12	9	6	TM0TGL/PA6	I/O	○		○		○					TM0TGL
13	10	7	TK0/PB0	I/O	○		○		○			○		
14	11	8	TK1/PB1	I/O	○		○		○			○		
15	12	9	TK2/PB2	I/O	○		○		○			○		
16	13	10	TK3/PB3	I/O	○		○		○			○		
17	14	11	PB4/TK4	I/O	○		○		○			○		
18	15	12	PB5/TK5	I/O	○		○		○			○		
19	16	13	PB6/TK6	I/O	○		○		○			○		
20	17	14	PB7/TK7	I/O	○		○		○			○		
21	18	15	PD0/TK8/IN0-/TM1TGL	I/O	○		○		○			○		TM1TGL/CMP
22	19	16	PD1/TK9/IN+	I/O	○		○		○			○		CMP
23	20	17	PD2/TK10/CMPO	I/O	○		○		○			○		CMP
24	21	18	PD3/TK11/IN1-	I/O	○		○		○			○		CMP
25	22	19	PD4/TK12	I/O	○		○		○			○		
26	23	20	PD5/TK13/TCOUT	I/O	○		○		○			○		TCOUT
27	24	-	PD6/TK14/PWM0	I/O	○		○		○		○	○		
28	25	-	PD7/PWMA	I/O	○		○		○		○			
29	26	21	PA3/Xout	I/O	○		○		○					
30	27	22	PA4/Xin/Xrc	I/O	○		○		○					
31	28	23	PA7/nRESET/VPP	I/O	○									nRESET
32	-	-	PE0	I/O	○		○		○					

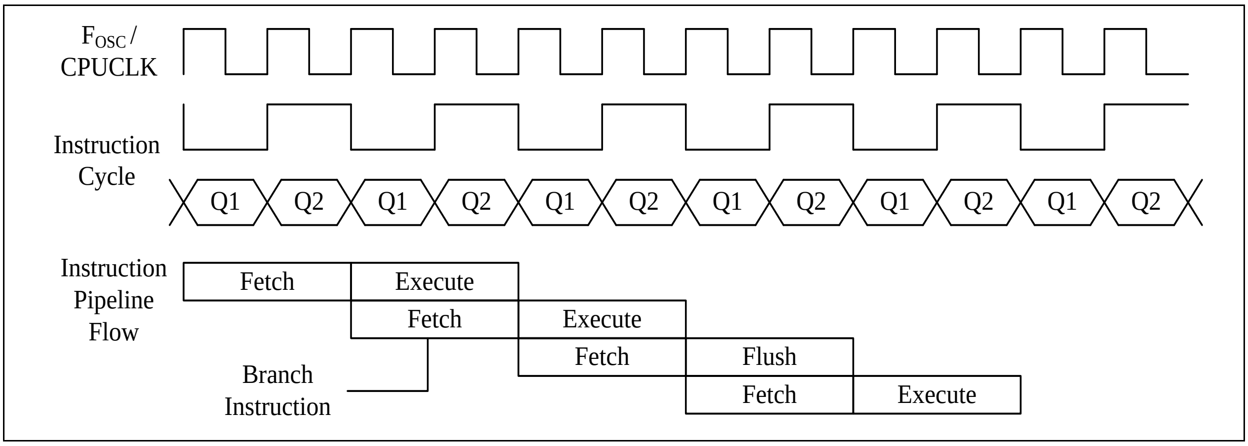
Symbol：P.P. = Push-Pull Output
P.O.D. = Pseudo Open Drain
O.D. = Open Drain
SYS = by SYSCFG bit

FUNCTIONAL DESCRIPTION

1. CPU Core

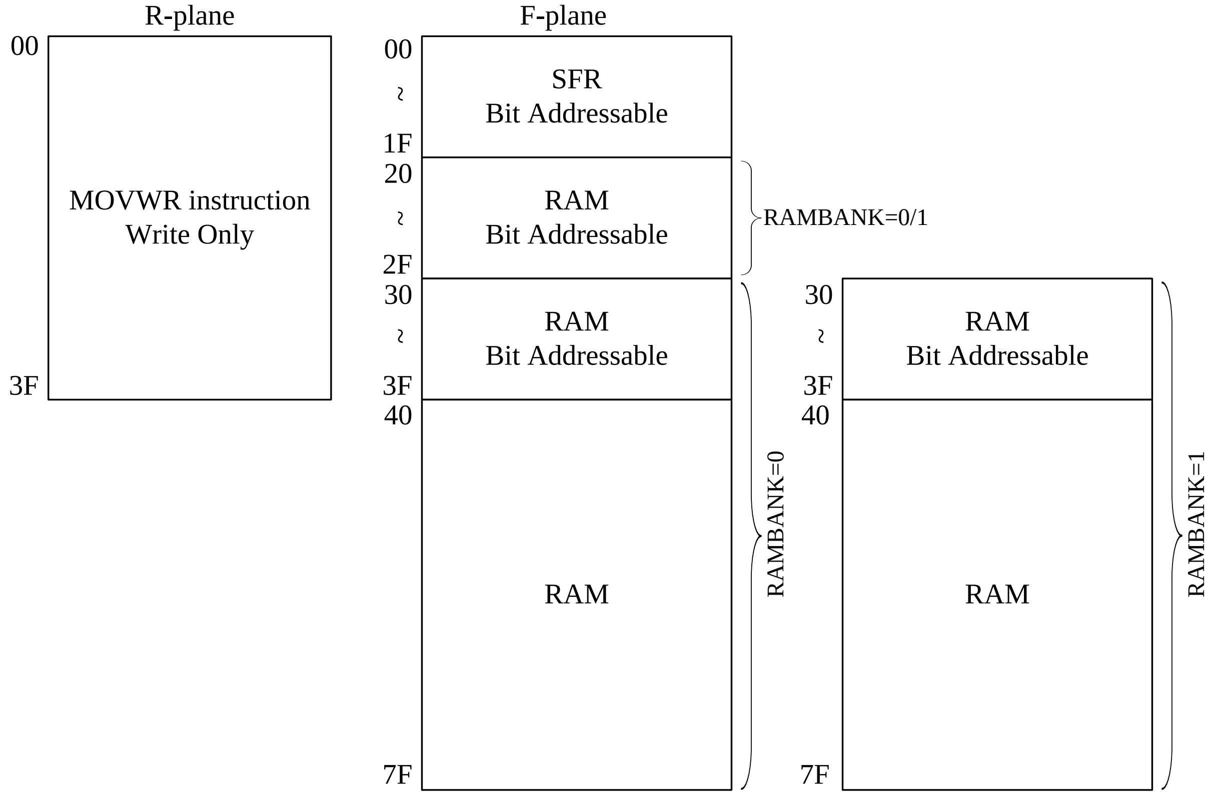
1.1 Clock Scheme and Instruction Cycle

The system clock is internally divided by two to generate Q1 state and Q2 state for each instruction cycle. The Programming Counter (PC) is updated at Q1 and the instruction is fetched from program ROM and latched into the instruction register in Q2. It is then decoded and executed during the following Q1-Q2 cycle. Branch instructions take two cycles since the fetch instruction is 'flushed' from the pipeline, while the new instruction is being fetched and then executed.



1.2 Addressing Mode

There are two Data Memory Planes in CPU, R-Plane and F-Plane. The registers in R-Plane are write-only. The “MOVWR” instruction copy the W-register’s content to R-Plane registers by direct addressing mode. The lower locations of F-Plane are reserved for the SFR. Above the SFR is General Purpose Data Memory, implemented as static RAM. F-Plane can be addressed directly or indirectly. Indirect Addressing is made by INDF register. The INDF register is not a physical register. Addressing INDF actually addresses the register whose address is contained in the FSR register (FSR is a pointer). The first half of F-Plane is bit-addressable, while the second half of F-Plane is not bit-addressable.



1.3 Programming Counter (PC) and Stack

The Programming Counter is 12-bit wide capable of addressing a 4K x 14 OTP ROM. As a program instruction is executed, the PC will contain the address of the next program instruction to be executed. The PC value is normally increased by one except the followings. The Reset Vector (000h) and the Interrupt Vector (001h) are provided for PC initialization and Interrupt. For CALL/GOTO instructions, PC loads 12 bits address from instruction word. For RET/RETI/RETLW instructions, PC retrieves its content from the top level STACK. For the other instructions updating PC [7:0], the PC [11:8] keeps unchanged. The STACK is 12-bit wide and 6-level in depth. The CALL instruction and hardware interrupt will push STACK level in order. While the RET/RETI/RETLW instruction pops the STACK level in order.

1.4 ALU and Working (W) Register

The ALU is 8-bit wide and capable of addition, subtraction, shift and logical operations. In two-operand instructions, typically one operand is the W register, which is an 8-bit non-addressable register used for ALU operations. The other operand is either a file register or an immediate constant. In single operand instructions, the operand is either W register or a file register. Depending on the instruction executed, the ALU may affect the values of Carry (C), Digit Carry (DC), and Zero (Z) Flags in the STATUS register. The C and DC flags operate as a /Borrow and /Digit Borrow, respectively, in subtraction.

Note: /Borrow represents inverted of Borrow register.

/Digit Borrow represents inverted of Digit Borrow register.

1.5 STATUS Register

This register contains the arithmetic status of ALU and the reset status. The STATUS register can be the destination for any instruction, as with any other register. If the STATUS register is the destination for an instruction that affects the Z, DC or C bits, then the write to these three bits is disabled. These bits are set or cleared according to the device logic. It is recommended, therefore, that only BCF, BSF and MOVWF instructions are used to alter the STATUS register because these instructions do not affect those bits.

STATUS	**Bit 7**	**Bit 6**	**Bit 5**	**Bit 4**	**Bit 3**	**Bit 2**	**Bit 1**	**Bit 0**
Reset Value	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**
R/W	**R/W**	**R/W**	**R/W**	**R**	**R**	**R/W**	**R/W**	**R/W**

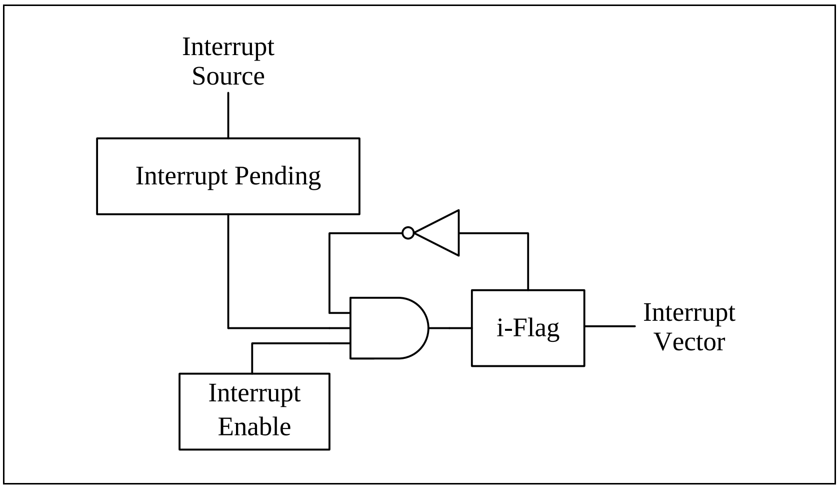
<table>
<tr><td>Bit</td><td colspan="2">Description</td></tr>
<tr><td>7</td><td colspan="2">General Purpose Bit</td></tr>
<tr><td>6</td><td colspan="2">General Purpose Bit</td></tr>
<tr><td>5</td><td colspan="2">**RAMBANK**</td></tr>
<tr><td>4</td><td colspan="2">**TO**: Time Out
0: after Power On Reset, LVR Reset, or CLRWDT/SLEEP instruction
1: WDT time out occurs</td></tr>
<tr><td>3</td><td colspan="2">**PD**: Power Down
0: after Power On Reset, LVR Reset, or CLRWDT instruction
1: after SLEEP instruction</td></tr>
<tr><td>2</td><td colspan="2">**Z**: Zero Flag
0: the result of a logic operation is not zero
1: the result of a logic operation is zero</td></tr>
<tr><td rowspan="3">1</td><td colspan="2">**DC**: Decimal Carry Flag or Decimal /Borrow Flag</td></tr>
<tr><td>ADD instruction</td><td>SUB instruction</td></tr>
<tr><td>1: a carry from the low nibble bits of the result occurs
0: no carry</td><td>1: no borrow
0: a borrow from the low nibble bits of the result occurs</td></tr>
<tr><td rowspan="3">0</td><td colspan="2">**C**: Carry Flag or /Borrow Flag</td></tr>
<tr><td>ADD instruction</td><td>SUB instruction</td></tr>
<tr><td>1: a carry occurs from the MSB
0: no carry</td><td>1: no borrow
0: a borrow occurs from the MSB</td></tr>
</table>

1.6 Interrupt

This device has 1 level, 1 vector and eight interrupt sources. Each interrupt source has its own enable control bit. An interrupt event will set its individual pending flag; no matter its interrupt enable control bit is 0 or 1. Because TM57PE40 has only 1 vector, there is not an interrupt priority register. The interrupt priority is determined by F/W.

If the corresponding interrupt enable bit has been set (INTE), it would trigger CPU to service the interrupt. CPU accepts interrupt in the end of current executed instruction cycle. In the mean while, a “CALL 001” instruction is inserted to CPU, and i-flag is set to prevent recursive interrupt nesting.

The i-flag is cleared in the instruction after the “RETI” instruction. That is, at least one instruction in main program is executed before service the pending interrupt. The interrupt event is level triggered. F/W must clear the interrupt event register while serving the interrupt routine.



2. Chip Operation Mode

2.1 Reset

This device can be RESET in four ways.

- Power-On-Reset
- Low Voltage Reset (LVR)
- External Pin Reset (PA7)
- Watchdog Reset (WDT)

After Power-On-Reset, all system and peripheral control registers are then set to their default hardware Reset values. The clock source, LVR level and chip operation mode are selected by the SYSCFG register value. The Low Voltage Reset features static reset when supply voltage is below a threshold level. There are three threshold levels can be selected. The LVR's operation mode is defined by the SYSCFG register. There are three voltage selections for the LVR threshold level. LVR32 is suitable for application with V_{DD} is more than 3.6V, LVR23 is suitable for application with V_{DD} is more than 2.7V, and LVR15 is suitable for application with V_{DD} is more than 1.9V. If operating frequency is faster than 16 MHz, choose LVR 3.2V is recommended.

See the following LVR Selection Table; user must also consider the lowest operating voltage of operating frequency.

LVR Selection Table:

LVR Threshold Level	Consider the operating voltage to choose LVR
LVR32	5.5V > VDD > 3.6V
LVR23	VDD is wide voltage range, more or less than 3.6V, more than 2.7V
LVR15	VDD is wide voltage range, more or less than 2.7V, more than 1.9V

The External Pin Reset and Watchdog Reset can be disabled or enabled by the SYSCFG register. These two resets also set all the control registers to their default reset value. The TO/PD flags is not affected by these resets.

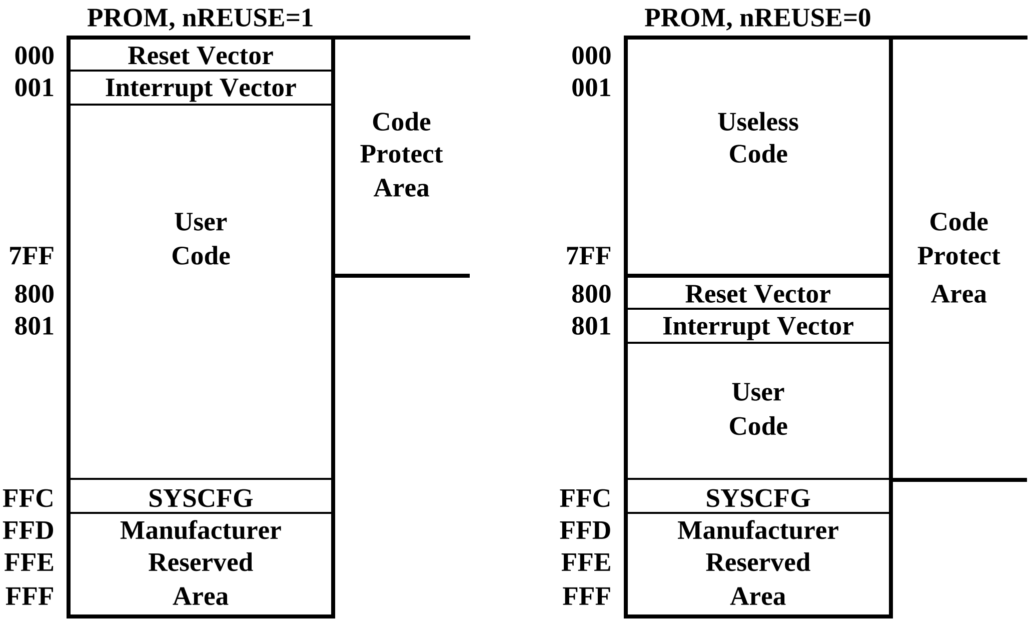
2.2 System Configuration Register (SYSCFG)

The System Configuration Register (SYSCFG) is located at ROM address FFCh. The SYSCFG determines the option for initial condition of MCU. It is written by PROM Writer only. User can select clock source, LVR threshold voltage and chip operation mode by SYSCFG register. The default value of SYSCFG is 14’b11_1111_111x_xxxx. The 13th bit of SYSCFG is code protection selection bit. If this bit is 0, the data in PROM will be protected, when user read PROM.

Bit	**13~0**	
Default Value	**11_1111_111x_xxxx**	
Bit	Description	
13	**nPROTECT**: Code protection selection	
	1	No protect
	0	Code protection
12	**nREUSE**: PROM Re-use control	
	1	Not Re-use
	0	Re-use
11-10	**LVR**: LV Reset Mode	
	11	LVR threshold is 1.5V, always enable
	10	LVR threshold is 1.5V, disable in sleep mode
	01	LVR threshold is 2.3V, always enable
	00	LVR threshold is 3.2V, always enable
9-8	**CLKS**: Fast Clock Source Selection	
	11	FXT (Fast Crystal) (1M~24 MHz)
	10	SXT (Slow Crystal, 32768 Hz)
	01	FIRC (Fast Internal RC, 2/4/8/16 MHz by FIRCSEL[1:0])
	00	XRC (External RC)
7	**XRESETE**: External pin Reset Enable	
	1	Enable External pin Reset (PA7 as reset pin)
	0	Disable External pin Reset (PA7 as input pin)
6	**WDTE**: WDT Reset Enable	
	1	Enable WDT Reset
	0	Disable WDT Reset
5	Test Mode, should be reserved as “1”	
4-0	**FIRCF**: Fast Internal RC Frequency adjustment control	

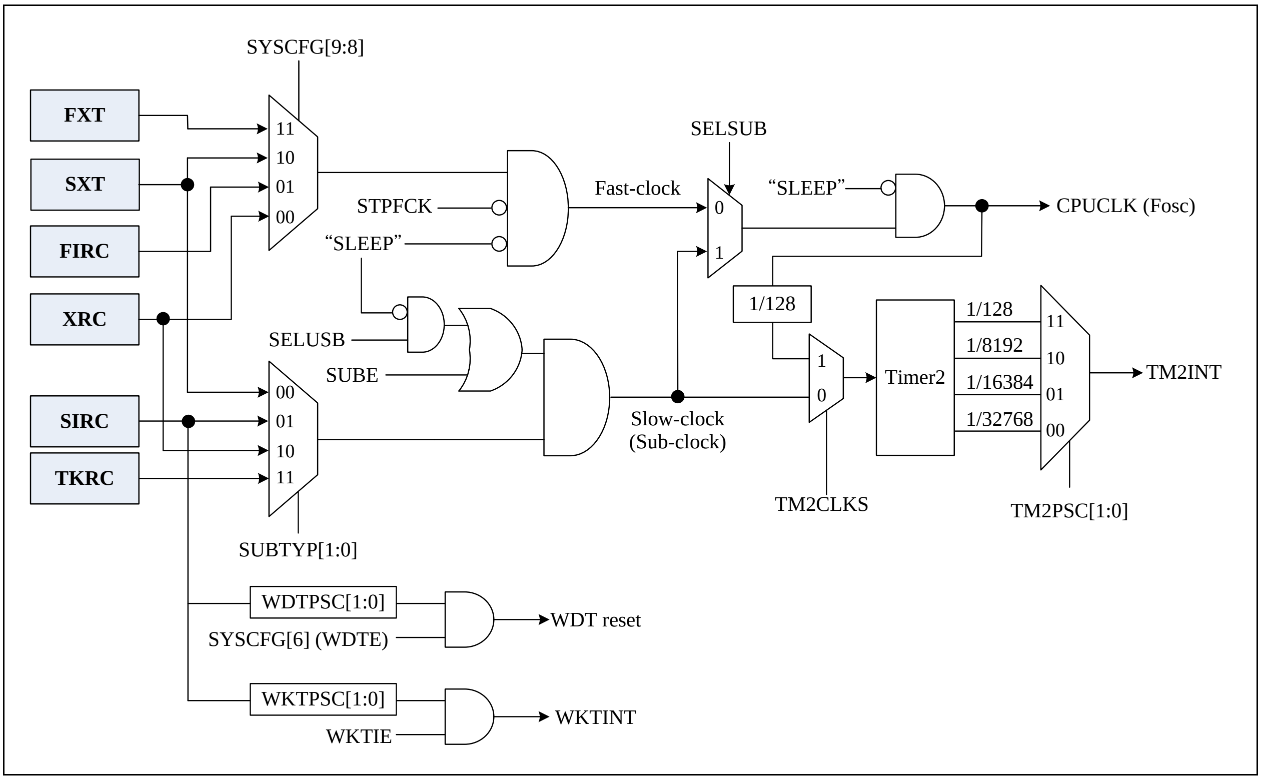
2.3 PROM Re-use ROM

The PROM of this device is 4K words. For some F/W program, the program size could be less than 2K words. To fully utilize the PROM, the device allows users to reuse the PROM. This feature is named as Two Time Programmable (TTP) ROM. While the first half of PROM is occupied by a useless program code and the second half of the PROM remains blank, users can re-write the PROM with the updated program code into the second half of the PROM. In the Re-use mode, the Reset Vector and Interrupt Vector are re-allocated at the beginning of the PROM's second half by the Assembly Compiler. Users simply choose the "REUSE" option in the ICE tool interface, and then the Compiler will move the object code to proper location. That is, the user's program still has reset vector at address 000h, but the compiled object code has reset vector at 200h. In the SYSCFG, if nPROTECT=0 and nREUSE=1, the Code protection area is first half of PROM. This allows the Writer tool to write then verify the Code during the Re-use Code programming. After the Re-use Code being written into the PROM's second half, user should write "nREUSE" control bit to "0". In the mean while, the Code protection area becomes the whole PROM except the Reserved Area.



2.4 Dual System Clock

TM57PE40 is designed with dual-clock system. There are six kinds of clock source, FXT (Fast Crystal) Clock, SXT (Slow Crystal) Clock, XRC (External RC) Clock, SIRC (Slow Internal RC) Clock, FIRC (Fast Internal RC) and TKRC (Touch Key Clock). Each clock source can be applied to CPU kernel as system clock. When in IDLE mode, only Slow-clock can be configured to keep oscillating to provide clock source to Timer2 block. Refer to the figure below.



Clock Scheme Block Diagram

FAST Mode

After power on or reset, TM57PE40 enters FAST mode. In FAST mode, TM57PE40 can select FXT, XRC or FIRC as its CPU clock by SYSCFG[9:8] setting. Besides, firmware can also enable or disable the Slow-clock for the Timer2 system operating.

In this mode, the program is executed using Fast-clock as CPU clock (CPUCLK). The Timer0, Timer1, PWM0, PWMA blocks are also driven by Fast-clock. Timer2 can also be driven by Fast-clock by setting TM2CLKS=1 and SELSUB=0.

SLOW Mode

In SLOW mode, TM57PE40 can select SXT, XRC, SIRC or TKRC as its CPU clock by R-Plane control register (SUBTYP[1:0]). In this mode, the Fast-clock is stopped and Slow-clock is enabled for power saving. All peripheral blocks (Timer0, Timer1, PWM0, PWMA, etc...) clock sources are Slow-clock in the SLOW mode.

IDLE Mode

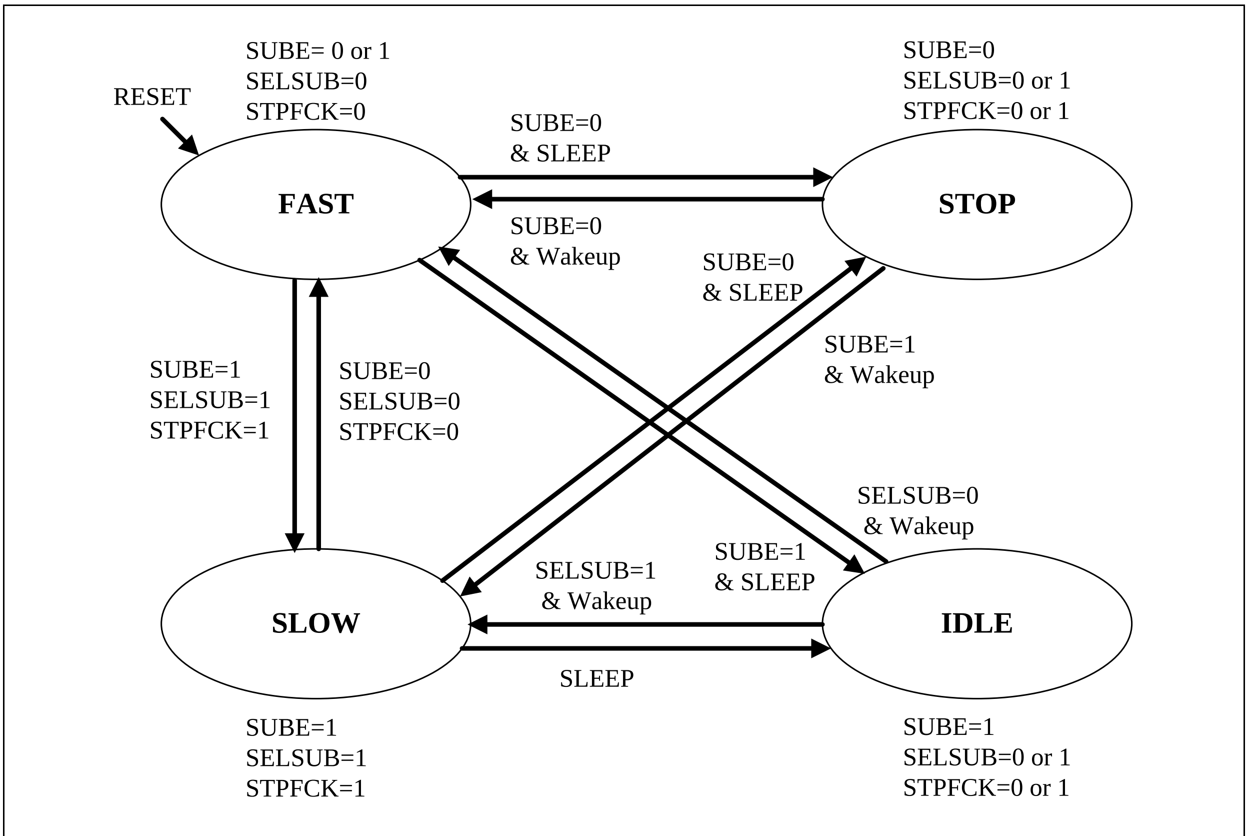
If Slow-clock is enabled and TM2CLKS=0 before executing the SLEEP instruction, the TM57PE40 enters the IDLE mode. In this mode, the Slow-clock will continue running to provide clock to Timer2 block. CPU stop fetching code and all blocks are stop except Timer2 related circuits.

STOP Mode

If Slow-clock is disabled before executing the SLEEP instruction, every block is turned off and the TM57PE40 enters the STOP mode. STOP mode is similar to IDLE mode. The difference is all clock oscillators either Fast-clock or Slow-clock is power down and no clock is generated.

2.5 Dual System Clock Modes Switching

TM57PE40 is operated in one of four modes: FAST mode, SLOW mode, IDLE mode, and STOP mode.



Note:
- SLEEP denotes SLEEP instruction
- Wakeup denotes wake-up events, such as XINT, PBWAKUP, WKTINT or TM2INT.

CPU Operation Block Diagram

Mode	Oscillator	CPUCLK	Fast-clock	Slow-clock	TM0/TM1	TM2	Wakeup event
FAST	FIRC, FXT, SXT, XRC	Fast-clock	Run	by STPFCK	Run	Run	X
SLOW	SXT, XRC, SIRC,TKRC	Slow-clock	by SUBE	Run	Run	Run	X
IDLE	SXT, XRC, SIRC,TKRC	Stop	Stop	Run	Stop	Run	WKT/IO/PB/TM2
STOP	SIRC[(1)]	Stop	Stop	Stop	Stop	Stop	WKT/IO/PB

(1) if WDT or WKT function is enabled

- FAST mode switches to SLOW mode

FAST mode can be chosen by SYSCFG [9:8] when equals to 11(FXT), 00(XRC), or 01(FIRC). The following steps are suggested to be executed by order when FAST mode switches to SLOW mode:

(1) Enable Slow-clock (SUBE=1)
(2) Switch to Slow-clock (SELSUB=1)
(3) Stop Fast-clock (STPFCK=1)

- SLOW mode switches to FAST mode

SLOW mode can be enabled by SUBE bit and SELSUB bit in CLKCTRL register. The following steps are suggested to be executed by order when SLOW mode switches to FAST mode:

(1) Enable Fast-clock (STPFCK=0)
(2) Switch to Fast-clock (SELSUB=0)
(3) Stop Slow-clock (SUBE=0)

Note: Stop Slow-clock (SUBE=0) is optional. Slow-clock can keep oscillating to provide Timer2 counter block in FAST mode.

- IDLE mode Setting

The IDLE mode can be configured by following setting in order:

(1) Enable Slow-clock (SUBE=1)
(2) Switch Timer2 clock source to Slow-clock (TM2CLKS=0)
(3) Execute SLEEP instruction

IDLE mode can be waken up by XINT, PBWAKUP, WKT timer interrupt and Timer2 interrupt.

- STOP Mode Setting

The STOP mode can be configured by following setting in order:

(1) Stop Slow-clock (SUBE=0)
(2) Execute SLEEP instruction

STOP mode can be waken up by XINT, PBWAKUP and WKT timer interrupt.

PA3/PA4 IO setting notes in dual clock mode

Note: In Slow-clock mode PA3 and PA4 must enable internal pull-high. If Slow-clock select SXT or XRC mode, the PA3 and PA4 IO setting list as below

	Fast-clock	Slow-clock	PAD3	PAE3	nPAPU3	PAD4	PAE4	nPAPU4
1	FIRC	SXT	1	0	0	1	0	0
2	FIRC	XRC	※	※	※	※	0	0
3	FXT	SIRC	1	0	0	1	0	0
4	XRC	SIRC	※	※	※	※	0	0

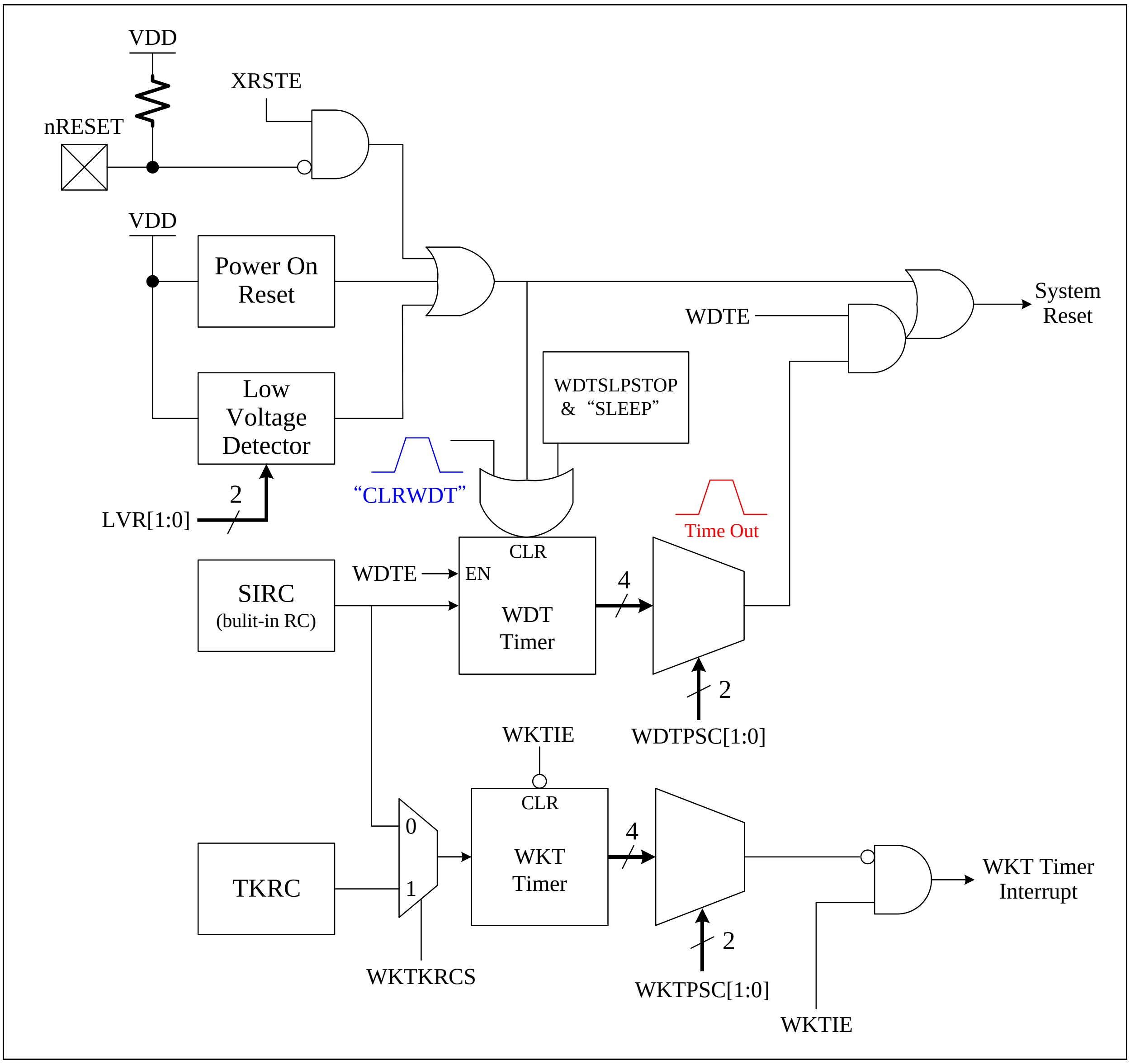
※：Don’t care

3. Peripheral Functional Block

3.1 Watchdog (WDT) / Wakeup (WKT) Timer

The WDT and WKT share the same internal RC Timer (SIRC). The overflow period of WDT, WKT can be selected by individual both four options (WDTPSC[1:0], WKTPSC[1:0]). The WDT timer is cleared by the CLRWDT instruction. If the Watchdog is enabled (WDTE=1), the WDT generates the chip reset signal. Set WDTSLPSTP (R0Eh.5) to ‘1’ can let WDT timer stop counting after executing SLEEP instruction, ie. WDTSLPSTP=0 WDT timer is always keep counting even if the SLEEP instruction is executed.

The WKT timer is an interval time, if WKT timer out will generate WKT Interrupt Flag (WKTIF). The WKT timer is cleared/stopped by WKTIE=0. Set WKTIE=1, the WKT timer will always count regardless at any CPU operating mode.



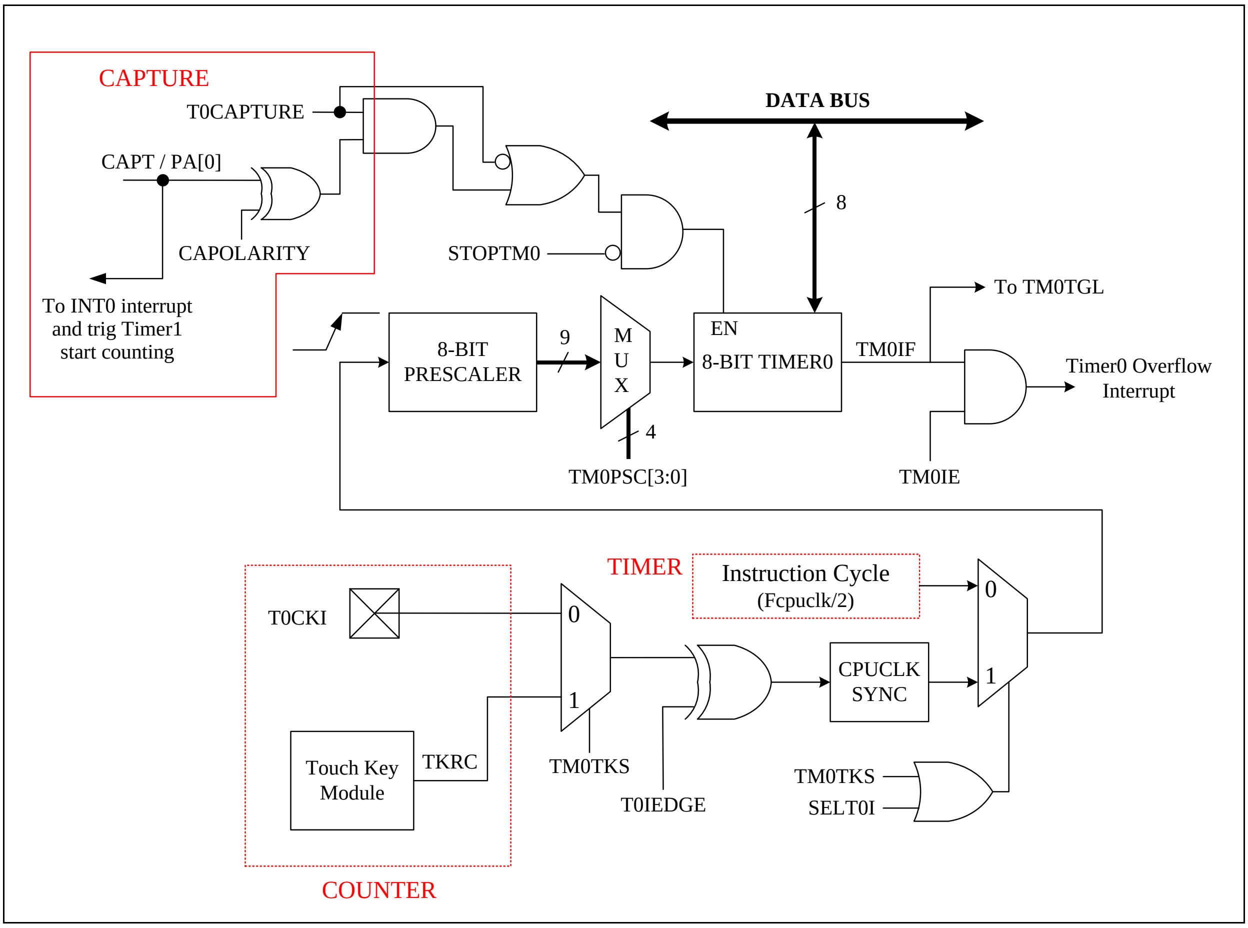
WDT/WKT Block Diagram

3.2 Timer0

The Timer0 is an 8-bit wide register of F-Plane 01h. It can be read or written as any other register of F-Plane. Besides, Timer0 increases itself periodically and automatically rolls over based on the pre-scaled clock source, which can be the instruction cycle or T0CKI (PE2) rising/falling input or Touch Key oscillating clock (TKRC) rising/falling. The Timer0 increase rate is determined by “Timer0 Pre-Scale” (TM0PSC) register in R-Plane. The Timer0 can generate interrupt flag (TM0IF) when it counts to rolls over if Timer0 Interrupt enable (TM0IE) is set. Timer0 can be stopped counting if the STOPTM0 bit is set. TM0TGL is an output signal that toggles when Timer0 overflows.

Timer0 can be configured as Capture mode. If T0CAPTURE bit is set to “1”, Timer0 will not count until the CAPT pin (i.e. PA0) is high level (CAPOLARITY=0) or low level (CAPOLARITY=1).

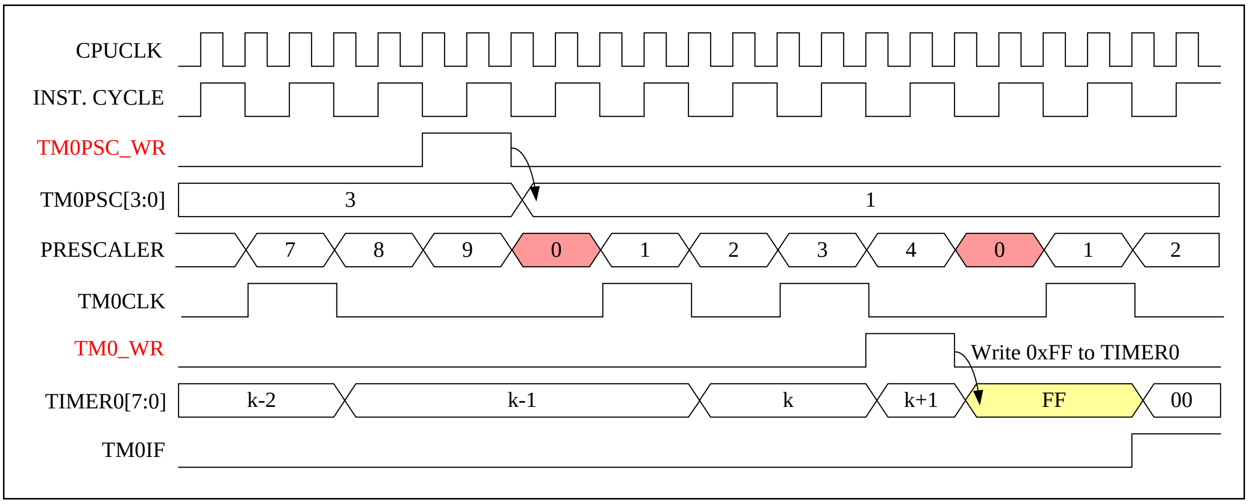
Timer0 can also be used to measure the pulse with and period capture on CAPT pin. This function needs the Timer1 and INT0 external interrupt. Software control details will be discussed step by step.



Timer0 Block Diagram

The following timing diagram describes the Timer0 works in pure timer mode.

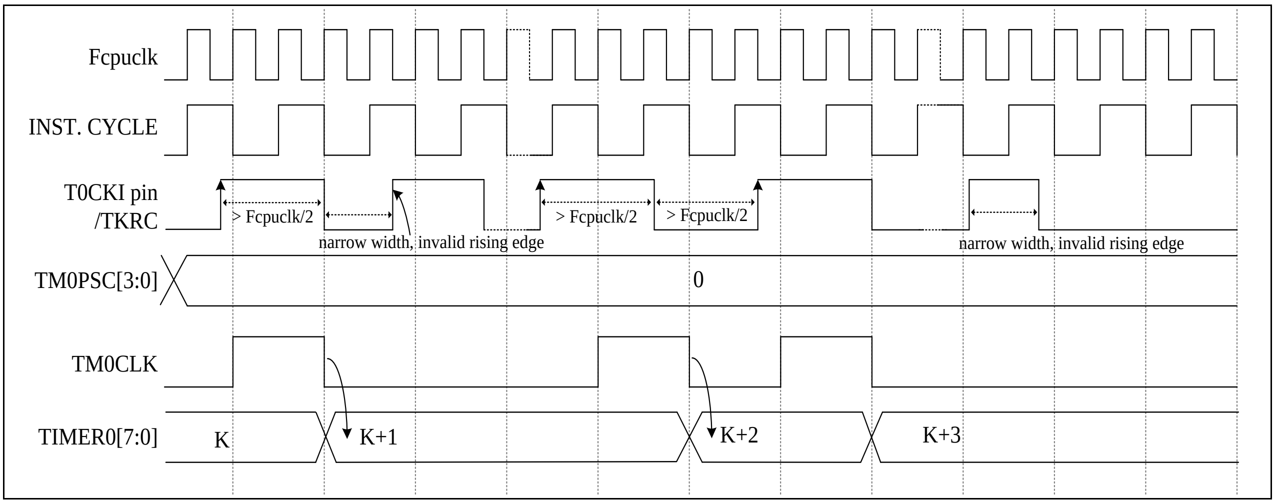
When the Timer0 prescaler (TM0PSC) is written, the internal 8-bit prescaler will be cleared to 0 to make the counting period correct at the first Timer0 count. TM0CLK is the internal signal that causes the Timer0 to increase by 1 at the end of TM0CLK. TM0WR is also the internal signal that indicates the Timer0 is directly written by instruction; meanwhile, the internal 8-bit prescaler will be cleared. When Timer0 counts from FFh to 00h, TM0IF (Timer0 Interrupt Flag) will be set to 1 and generate interrupt if TM0IE (Timer0 Interrupt Enable) is set.



Timer0 works in Timer mode

The following timing diagram describes the Timer0 works in counter mode.

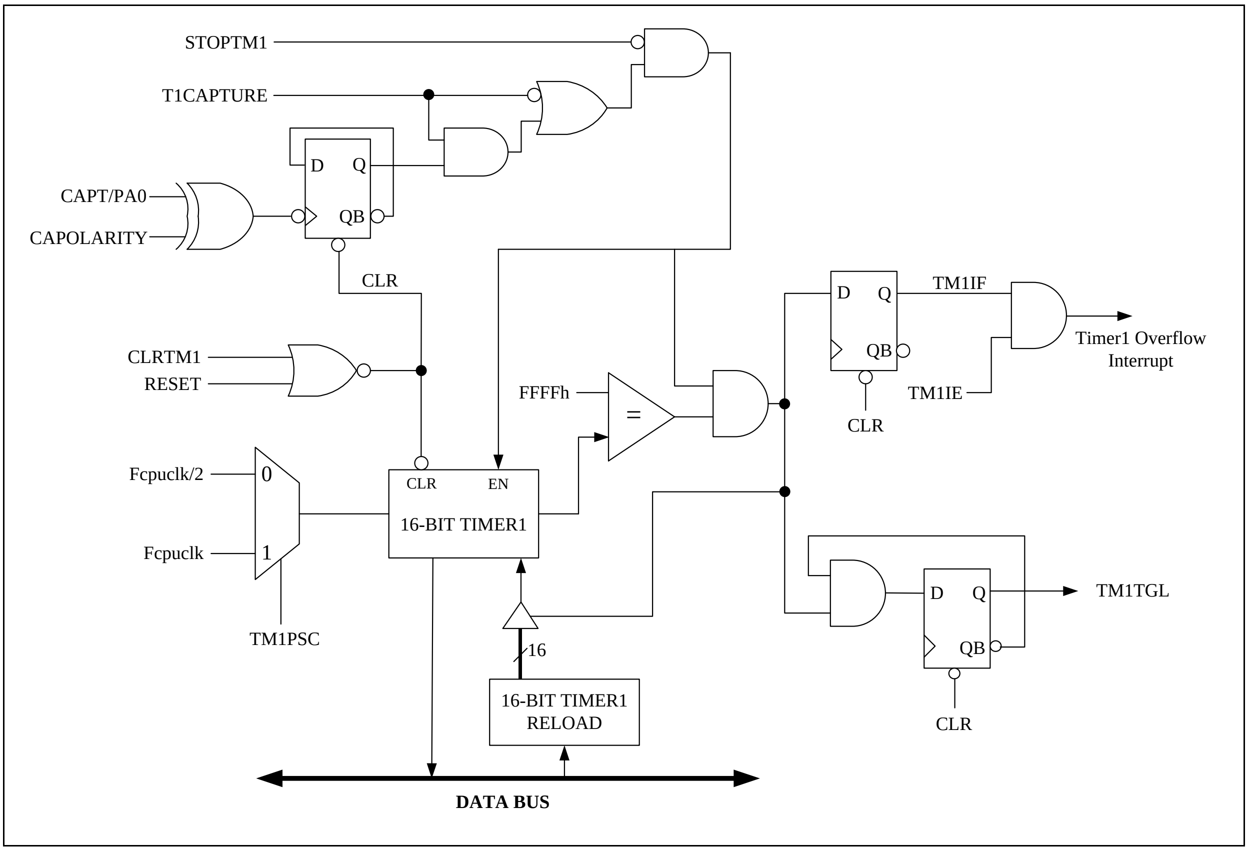
If SELT0I=1 then the Timer0 counter source clock is from T0CKI pin (TM0TKS=0) or Touch Key module (TM0TKS=1). T0CKI or TKRC signal is synchronized by instruction cycle (i.e. Fcpuclk/2), that means the high/low time durations of T0CKI/TKRC must be longer than one instruction cycle time to guarantee each T0CKI's/TKRC's change will be detected correctly by the synchronizer.



Timer0 works in Counter mode for T0CKI/TKRC (T0IEDGE=0)

3.3 Timer1

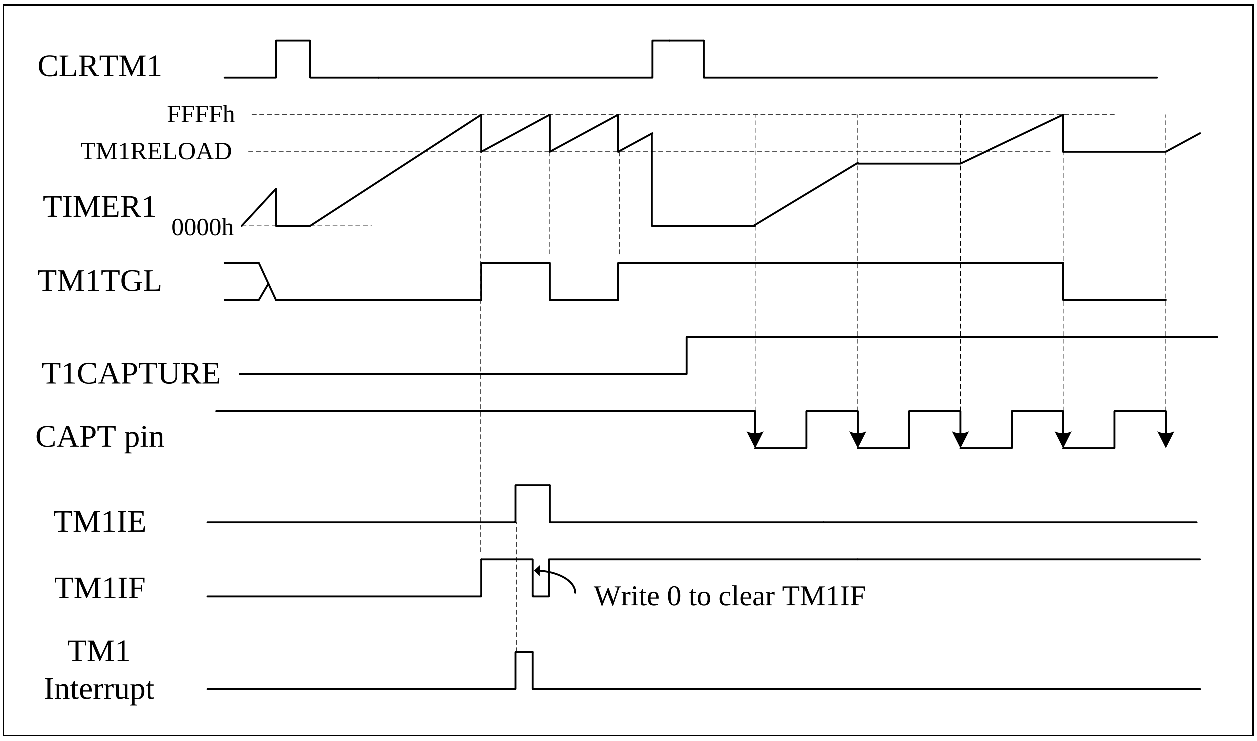
Timer1 is a 16-bit counter used as Capture/Timer mode with 16-bit auto-reload register. Timer1 can only be accessed by reading F-plane TM1H and TM1L. Writing TM1H and TM1L is actually writing to Timer1 reload registers. The clock sources of Timer1 are Fcpuclk and Fcpuclk/2, selected by TM1PSC. Setting the bit CLRTM1 will clear Timer1 and hold Timer1 on 0000h. Setting the STOPTM1 bit will stop Timer1 counting. TM1TGL is an output signal that toggles when Timer1 overflow.



Timer1 Block Diagram

Note that writing to TM1H and TM1L is actually writing to Timer1 reload register, while reading TM1H and TM1L is actually reading the Timer1 counter itself. That is, Timer1 counter and Timer1 reload register share two addresses (0ah, 0bh) of F-plane.

Timer1 can also works with Capture mode. When works in Capture mode, Timer1 will start counting when the CLRTM1 bit is cleared and the first falling edge of CAPT pin (if CAPOLARITY=0) is coming. When the 2nd falling edge of CAPT pin is coming, Timer1 stops counting and hold the value. When the 3rd falling edge of CAPT pin is coming, the Timer1 continue counting. The following figure shows the detail timing diagram.



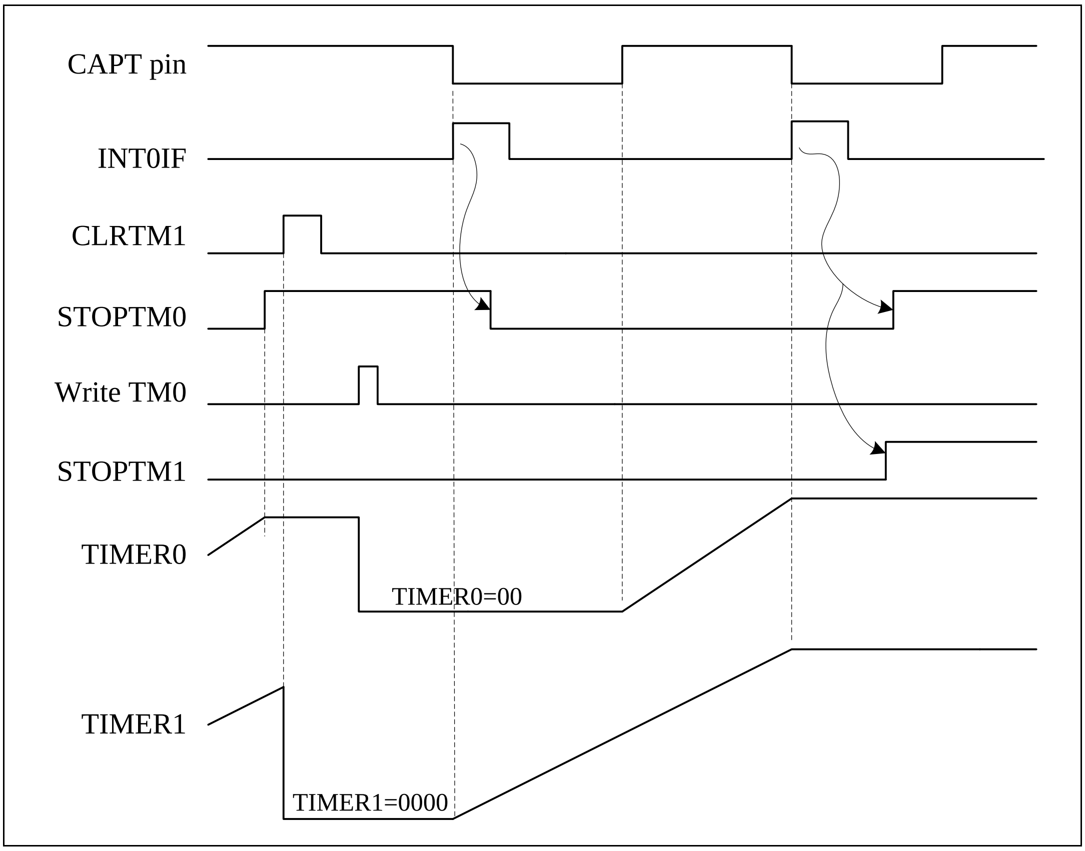
Timer1 works in Capture mode (CAPOLARITY=0, implies CAPT falling edge)

3.4 Timer0 and Timer1 Used for Pulse Width and Period Capture

Timer0 and Timer1 can cooperate to measure the signal period and duty cycle time. The key is multi-function of PA0 (CAPT, INT0). Suppose that:

- SELT0I=0 and TM0TKS=0, Timer0 prescaler increases per instruction cycle.
- T0CAPTURE=1, T1CAPTURE=1. Timer0 and Timer1 work in Capture mode.
- PA0 pin (CAPT pin) interrupts every falling edge.
- CAPOLARITY=0, **Timer1** starts/holds in turn when PA0 pin (CAPT pin) falling edge is coming. **Timer0** starts counting when PA0 pin (CAPT pin) is in logic ‘1’ level, and holds the Timer0 value when PA0 pin (CAPT pin) is in logic ‘0’ level.
- Timer1 is used to measure the signal period, Timer0 is used to measure the PA0 (CAPT pin) in logic ‘1’ time (i.e. the duty cycle of the signal).

The following figure shows how to use Timer0 and Timer1 to measure the PA0 (CAPT pin) signal’s period and duty cycle (CAPOLARITY=0).

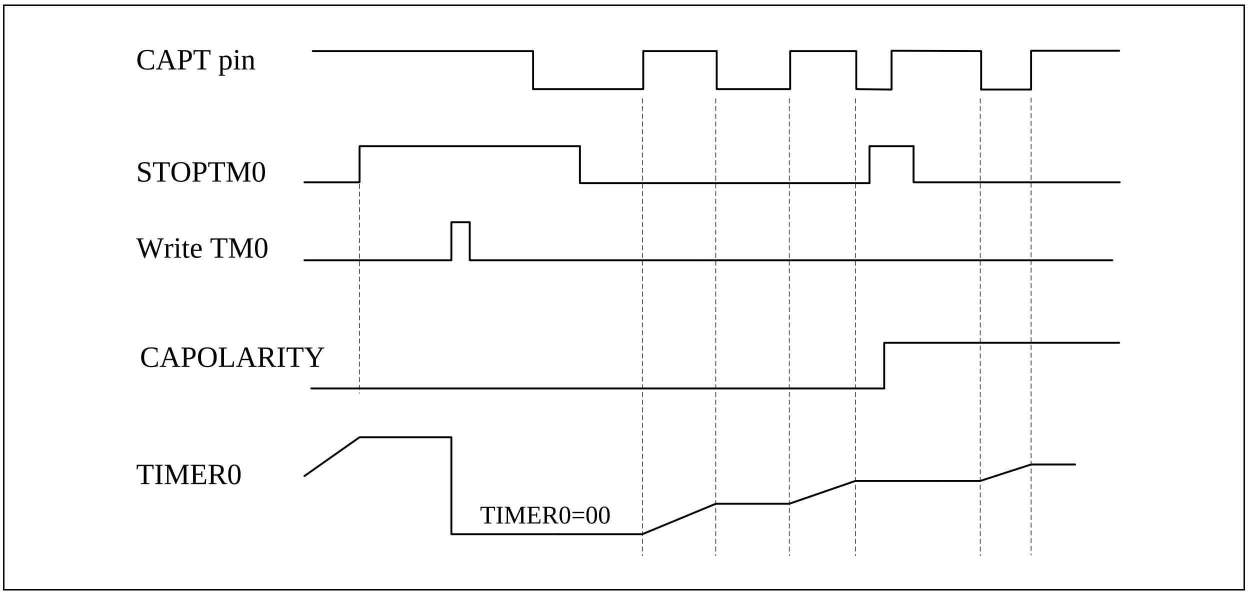


Timer0 and Timer1 are used to measure the signal on CAPT pin.

Follow the steps below to start measuring the CAPT pin's period and duty cycle.

1. Stop Timer0 by firmware (STOPTM0=1, Timer0 will be stopped and hold)
2. Clear Timer1 by firmware (CLRTM1=1)
3. Clear Timer0 by directly write 00h to Timer0 (Timer0 is still hold)
4. Once CAPT pin falling edge is coming, the Timer1 starts counting; meanwhile the PA0 interrupt is generated and STOPTM0 is cleared by the firmware. Now the Timer0 is ready to count when CAPT pin goes high)
5. CAPT pin rising edge is coming, Timer0 starts counting until the CAPT pin returns to 0 and holds the counting value. Timer1 also stops counting and holds the value.
6. PA0 interrupt is generated again, firmware stops Timer1 and Timer0 to read the period and duty cycle.

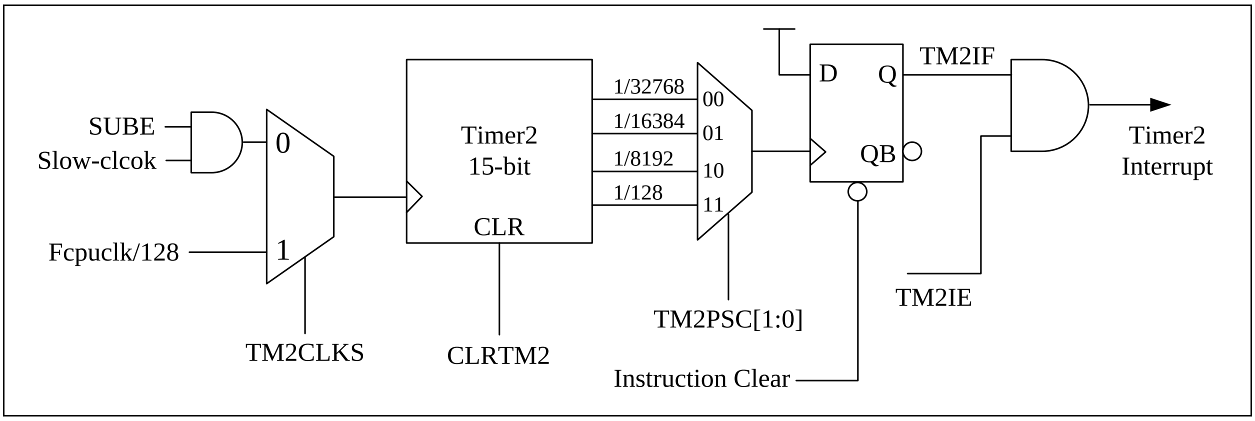
It is not necessary to use both Timer0 and Timer1. If only the duty cycle (CAPT high time) needs to be measured, there is no need to use Timer1 to measure the period. In such case, user can set the T0CAPTURE=1 and T1CAPTURE=0. Timer0 is counting up only when CAPT pin is '1'. Note that the internal prescaler will be kept to next Timer0 count, so it will not lose the counting accuracy.



Timer0 is used to measure the high (or low) time on CAPT pin

3.5 Timer2: 15-bit Timer

The Timer2 is a 15-bit counter and the clock sources are from either Fcpuclk/128 or Slow-clock. It is used to generate time base interrupt and Timer2 counter block clock. The Timer2 content cannot be read by instructions. It generates interrupt flag (TM2IF) with the clock divided by 32768/16384/8192/128 depends on TM2PSC[1:0] register bits. The following figure shows the block diagram of Timer2.



Timer2 Block Diagram

Example:

CPU operating procedure is from SLOW (SXT=32 KHz) switch to IDLE mode and executes the wake up interrupt subroutine per 0.5 sec.

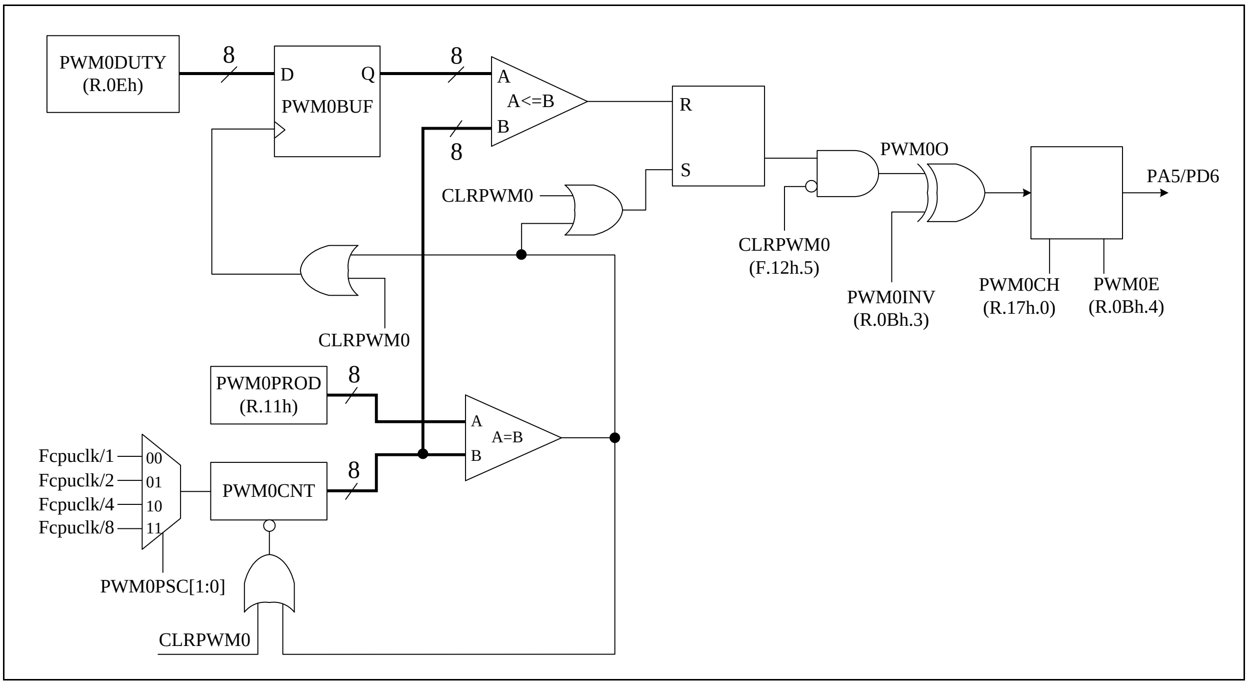
[CPU running at SLOW mode, STPFCK=1, SUBE=1, SELSUB=1, WDTE=0]

```
movlw    11000001    ; TM2CLKS=Slow-clock(SXT=32 KHz), TM2PSC=div16384
bsf      TM2IE       ; Timer2 interrupt enable
bcf      CLRTM2      ; Don’t clear timer2
sleep                ; enter IDLE mode, wait Timer2 interrupt happen
nop
goto     $-2         ;
```

3.6 PWM0: 8-bit PWM

The chip has a built-in 8-bit PWM generator. The source clock comes from Fcpuclk divided by 1, 2, 4, and 8. The PWM0 duty cycle can be changed by writing to PWM0DUTY, writing to PWM0DUTY will not change the current PWM0 duty until the current PWM0 period completes. When current PWM0 period is finish, the new value of PWM0DUTY will be updated to the PWM0BUF.

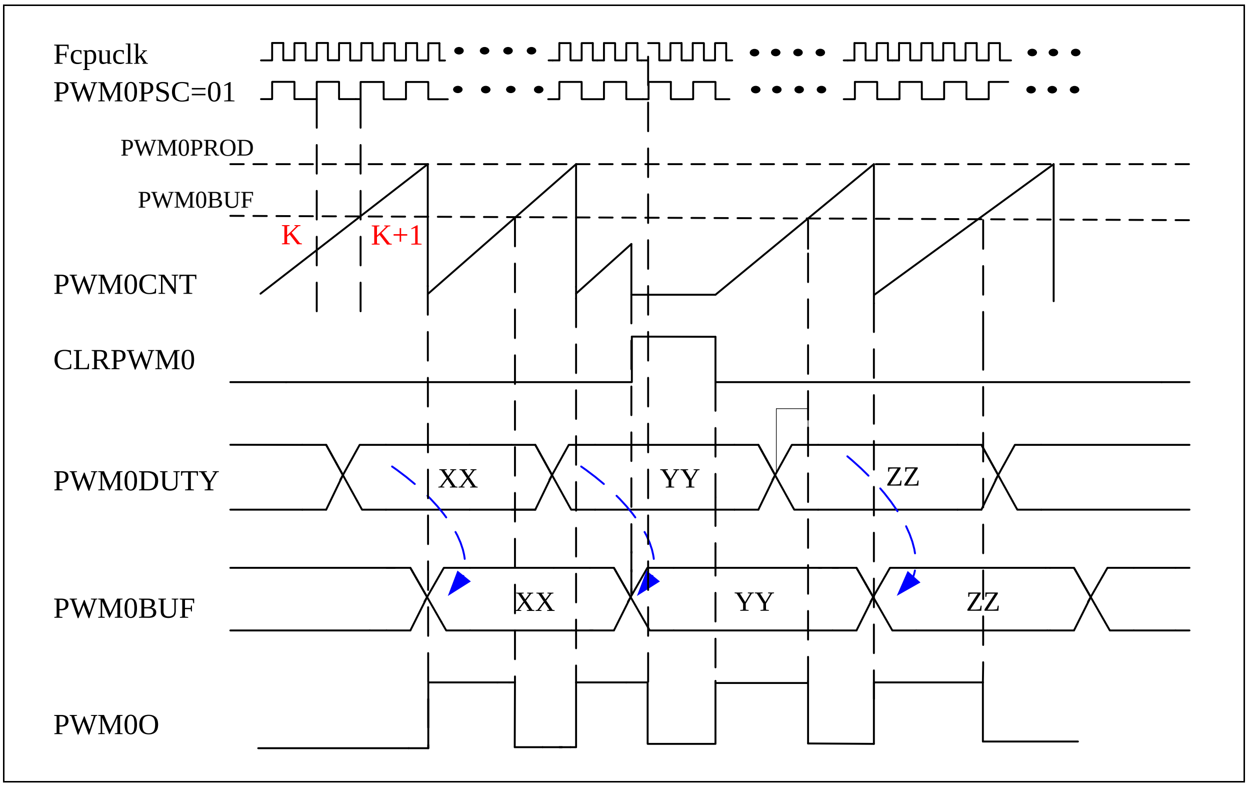
The PWM0INV can inverse the PWM0 output. PWM0 alternative output channel can be PA5 or PD6 by PWM0CH bit control, if PWM0E is set to 1. Setting the CLRPWM0 bit will clear the PWM0 counter and load the PWM0DUTY to PWM0BUF, CLRPWM0 bit must be cleared so that the PWM0 counter can count. The following figure shows the block diagram of PWM0.



PWM0 Block Diagram

The following figure shows the PWM0 waveforms. When CLRPWM0 bit is set to ‘1’, the PWM0 output is cleared to ‘0’ no matter what its current status is. Once the CLRPWM0 bit is cleared to ‘0’, the PWM0 output is set to ‘1’ to begin a new PWM cycle. PWM0 output will be ‘0’ when PWM0CNT greater than or equals to PWM0BUF. PWM0CNT keeps counting up when equals to PWM0PROD, the PWM0 output is set to ‘1’ again.

The PWM0 period can be set by writing period value to PWM0PROD register. Note that changing the PWM0PROD is immediately changing the PWM0PROD values in the Figure that is different from PWM0DUTY which has PWM0BUF to update the duty at the end of current period. The Programmer must pay attention to the current time to change PWM0PROD by observing the following figure. There is a digital comparator that compares the PWM0CNT and PWM0PROD, if PWM0CNT is larger than PWM0PROD after setting the PWM0PROD, a fault long PWM cycle will be generated because PWM0CNT must count to overflow then keep counting PWM0PROD to finish the cycle.



PWM0 duty/period calculation is as follows:

- The PWM0 output duty = PWM0DUTY / (PWM0PROD + 1)
  If PWM0DUTY = 80H, PWM0PROD = FFH, the PWM0 output duty will be 50%
- The PWM0 output frequency = (Fcpuclk) / (PWM0 Prescaler) / (PWM0PROD + 1)

PWM0PSC=2’b10, the PWM0 Prescaler = 4 (Fcpuclk divided by 4)

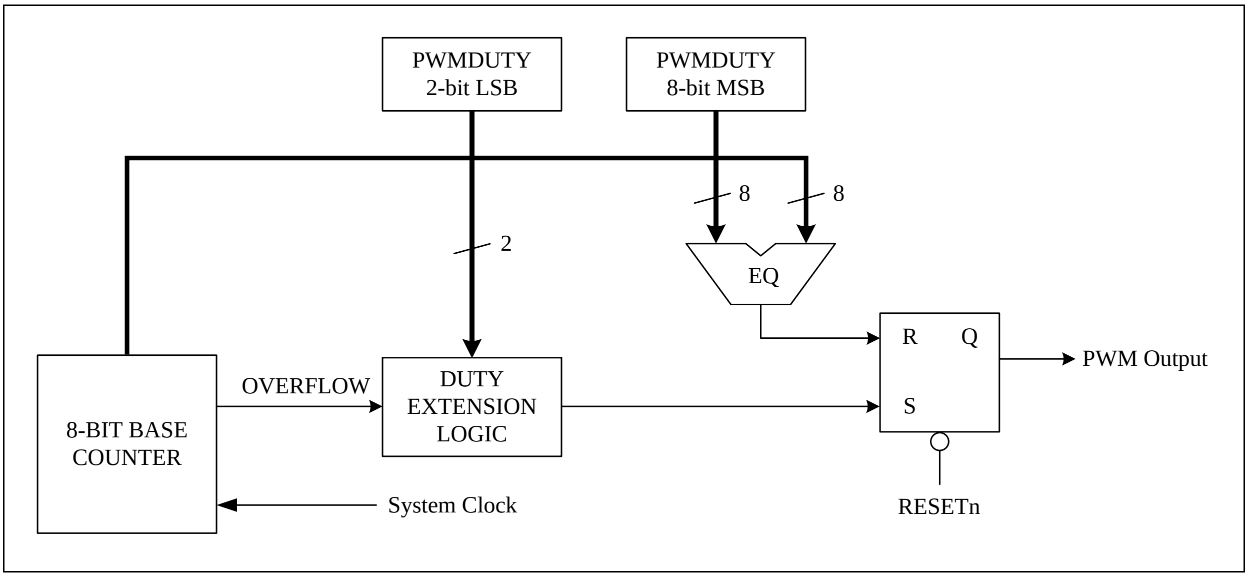
PWM0 period data (PWM0PROD) = 39

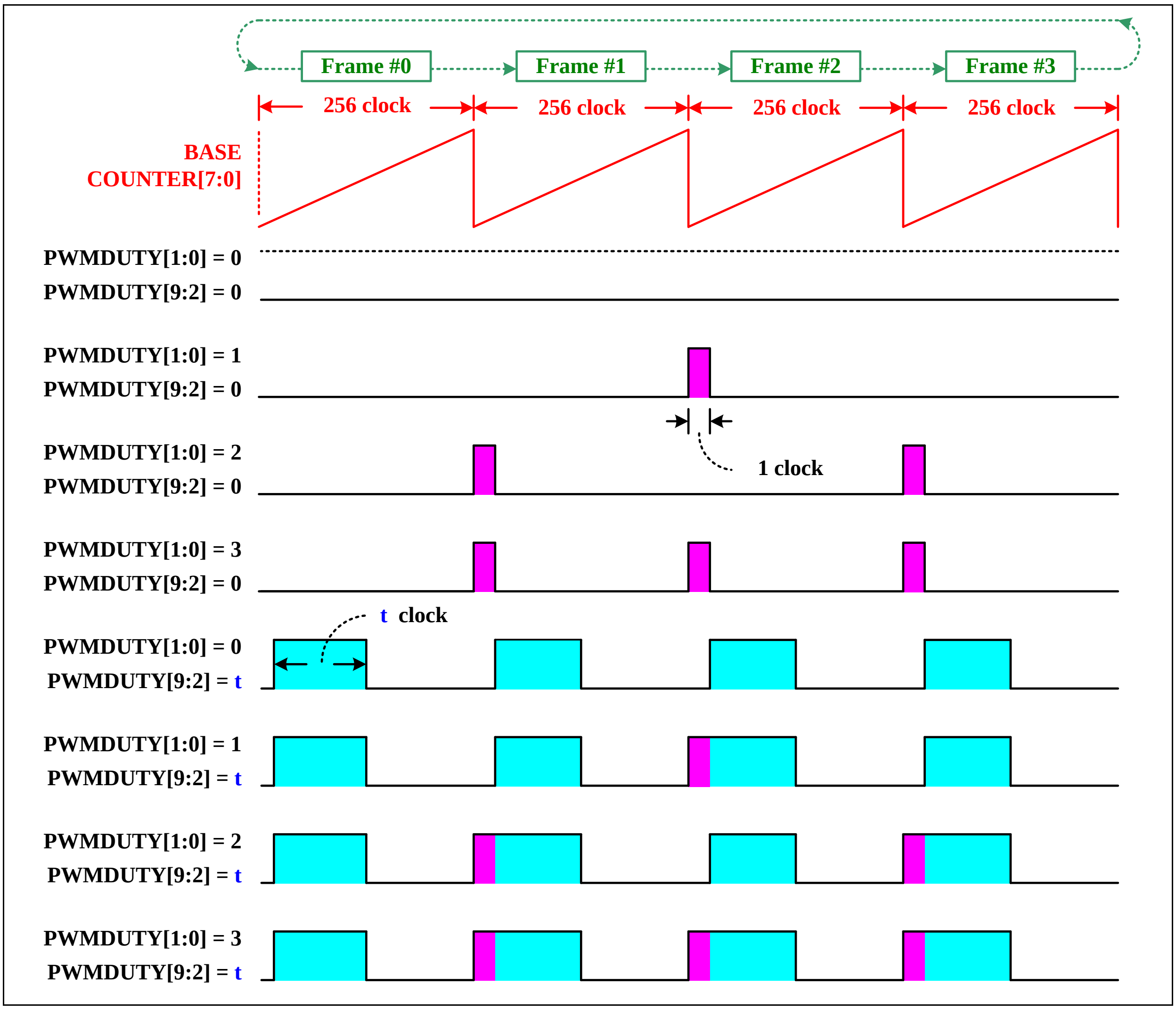
Fcpuclk=4 MHz

PWM0 output frequency = (4M)/(4)/(40) = 25000 Hz

3.7 PWMA: (8+2) bits PWM

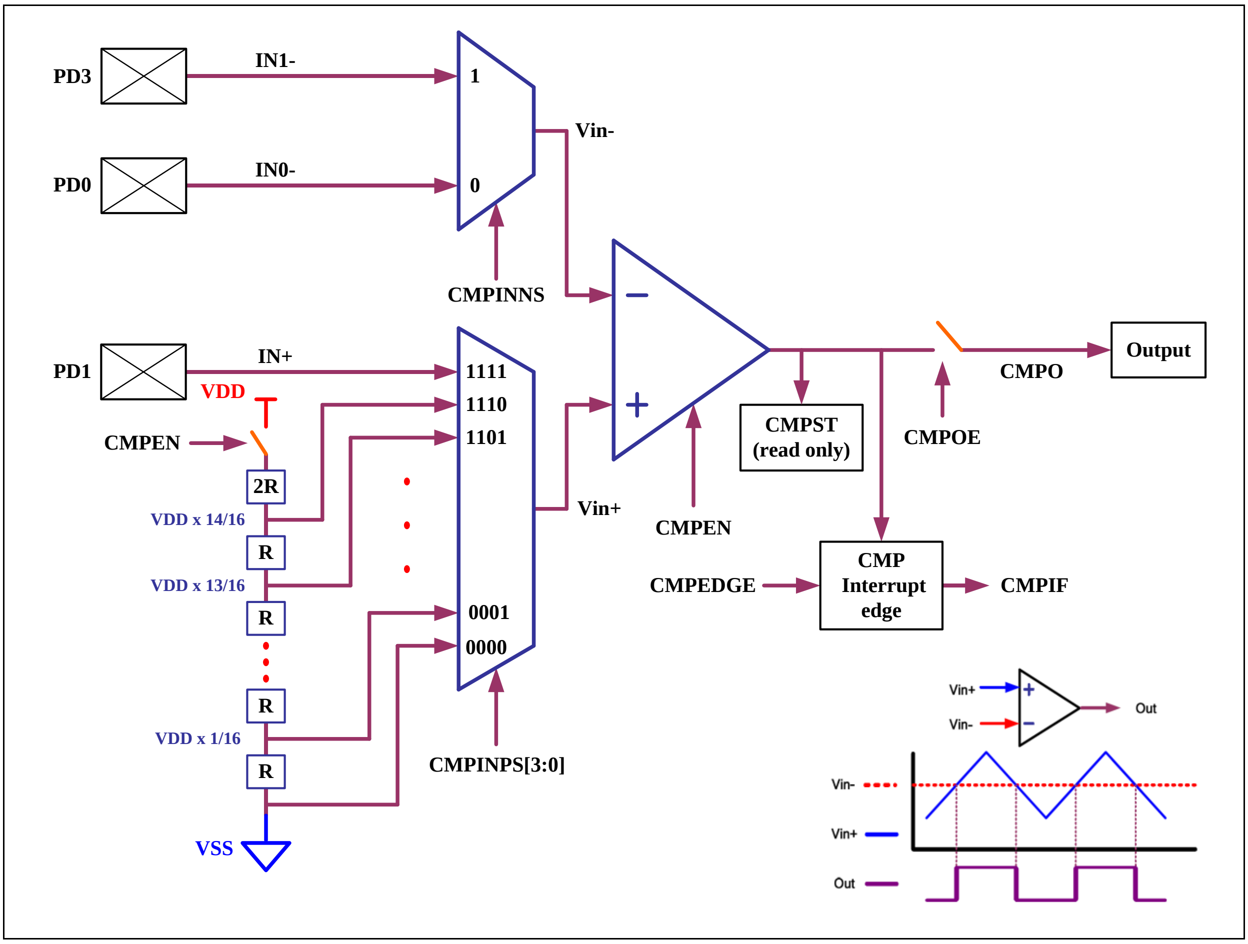
The PWM can generate fix frequency waveform with 1024 duty resolution based on System Clock (Fcpuclk). A spread LSB technique allows PWM to run its frequency at “System Clock divided by 256” instead of “System Clock divided by 1024”, which means the PWM is 4 times faster than normal. The advantage of higher PWM frequency is that the post RC filter can transform the PWM signal to more stable DC voltage level. The PWM output signal reset to low level whenever the 8-bit base counter matches the 8-bit MSB of PWM duty register (PWM0DUTY). When the base counter rolls over, the 2-bit LSB of PWM duty register decides whether to set the PWM output signal high immediately or set it high after one clock cycle delay.




Frame #0
Frame #1
Frame #2
Frame #3
256 clock
256 clock
256 clock
256 clock
BASE COUNTER[7:0]
PWMDUTY[1:0] = 0
PWMDUTY[9:2] = 0
PWMDUTY[1:0] = 1
PWMDUTY[9:2] = 0
1 clock
PWMDUTY[1:0] = 2
PWMDUTY[9:2] = 0
PWMDUTY[1:0] = 3
PWMDUTY[9:2] = 0
t clock
PWMDUTY[1:0] = 0
PWMDUTY[9:2] = t
PWMDUTY[1:0] = 1
PWMDUTY[9:2] = t
PWMDUTY[1:0] = 2
PWMDUTY[9:2] = t
PWMDUTY[1:0] = 3
PWMDUTY[9:2] = t

3.8 Analog Comparator

TM57PE40 includes an analog comparator. It can be enabled by CMPEN which is in (R10h.7). The analog comparator compares the input values on the positive pin Vin+ and negative pin Vin-. When the voltage on positive pin is higher than the voltage on the negative pin, the analog comparator out (CMPO) is set. The output status CMPST can be read from (F0Dh.0) or output to pin by setting CMPOE which in (R10h.5). The analog comparator can generate interrupt flag (CMPIF) when the output status changes. The user can select interrupt triggering on comparator output rise or fall. The input source of negative pin can be selected from IN0- or IN1- by CMPINNS. The analog comparator support internal reference voltage. The internal reference voltage provides the range of output voltage with 15 distinct levels. The range can be selected by CMPINPS[3:0]. A block diagram of the analog comparator is shown below.



Comparator Block Diagram

3.9 Touch Key

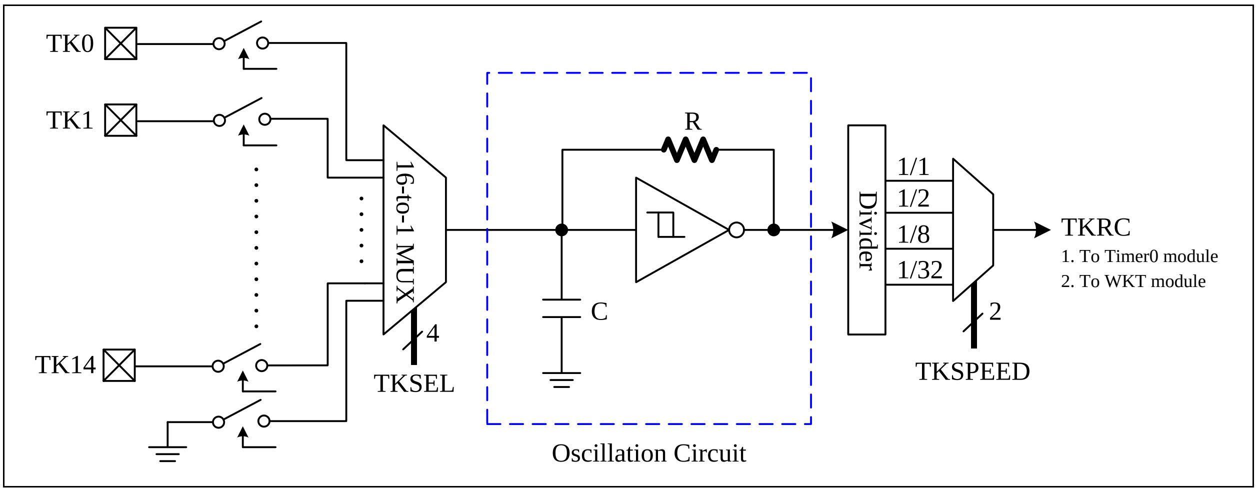
The Touch Key Module outputs the oscillation clock to both WKT module and Timer0 module. The TKRC to Timer0 is like T0I input. Touch Key module consists of a RC oscillator, 16-to-1 analog input select, TKSPEED control bits select the output of the frequency divider. The frequency divider divides the oscillation clock by 1, 2, 8, and 32. If WKTKRCS bit is 1, the divided clock will be sent to WKT module; whereas if TM0TKS bit is 1, the divided clock will be sent to Timer0 module. Timer0 counts at the rising or falling edge depends on T0IEDGE bit.

If the human finger tips close to the touch pad, the equivalent capacitance of C will be increased, that is, the oscillation frequency will be decreased.

Based on the above thesis, user program needs to observe what input channel causes the lowest Timer0 counting value in a fixed period of time, which channel of key is touched or the finger is just approaching.

To distinguish what channel counting value is the lowest, we need another Timer (Timer1, Timer2, or WKT Timer) to set up a proper interval of time that Timer0 will not count to overflow. Based on this fixed time interval, the user program switches the Touch Key channels one after another and finds the lowest value of Timer0, which is the key in touching or approaching.

The Touch Key Oscillation circuit structure is similar to WKT, therefore, if using WKT module to calculate touch key counts will obtain the best performances, independent both the temperature and voltage coefficient.

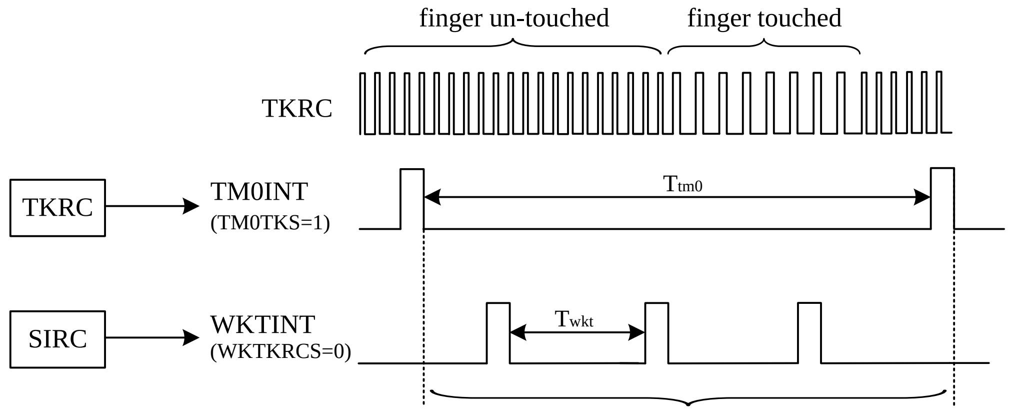


Touch Key Oscillator Block Diagram

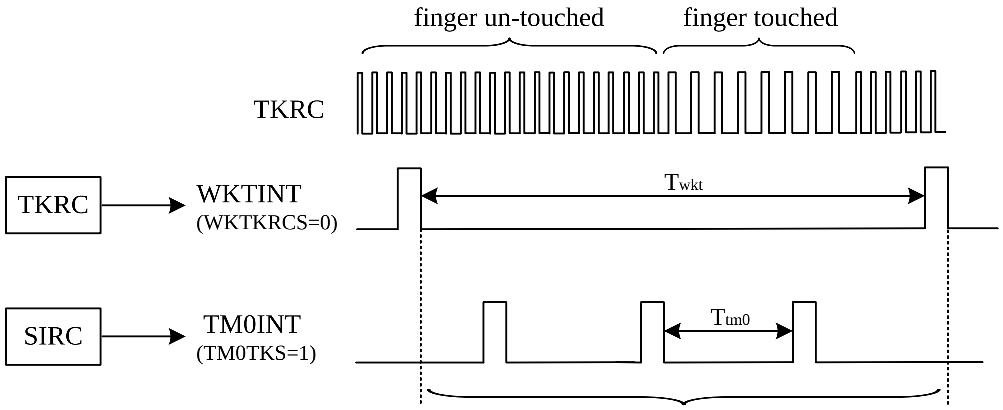
Example:

```
movlw    1000000b     ; disable TK7(PB7) pull up resistor
movwr    nPBPU        ; R09h (R-plane address 09h)
movlw    10000111b    ;WKTKRCS=1,  WKT timer clock source from TK7(PB7) pin.
                       TM0TKS=0
                      ; TKSPEED=2'b00 (fastest speed), TKSEL=4'b0111
movwr    TKCTL        ; R0Dh
movlw    01111111b    ; disable PB7 digital input.
movwr    PB_IE         ; R12h
```

3.9.1 Touch Key Using Timer0 and WKT Timer

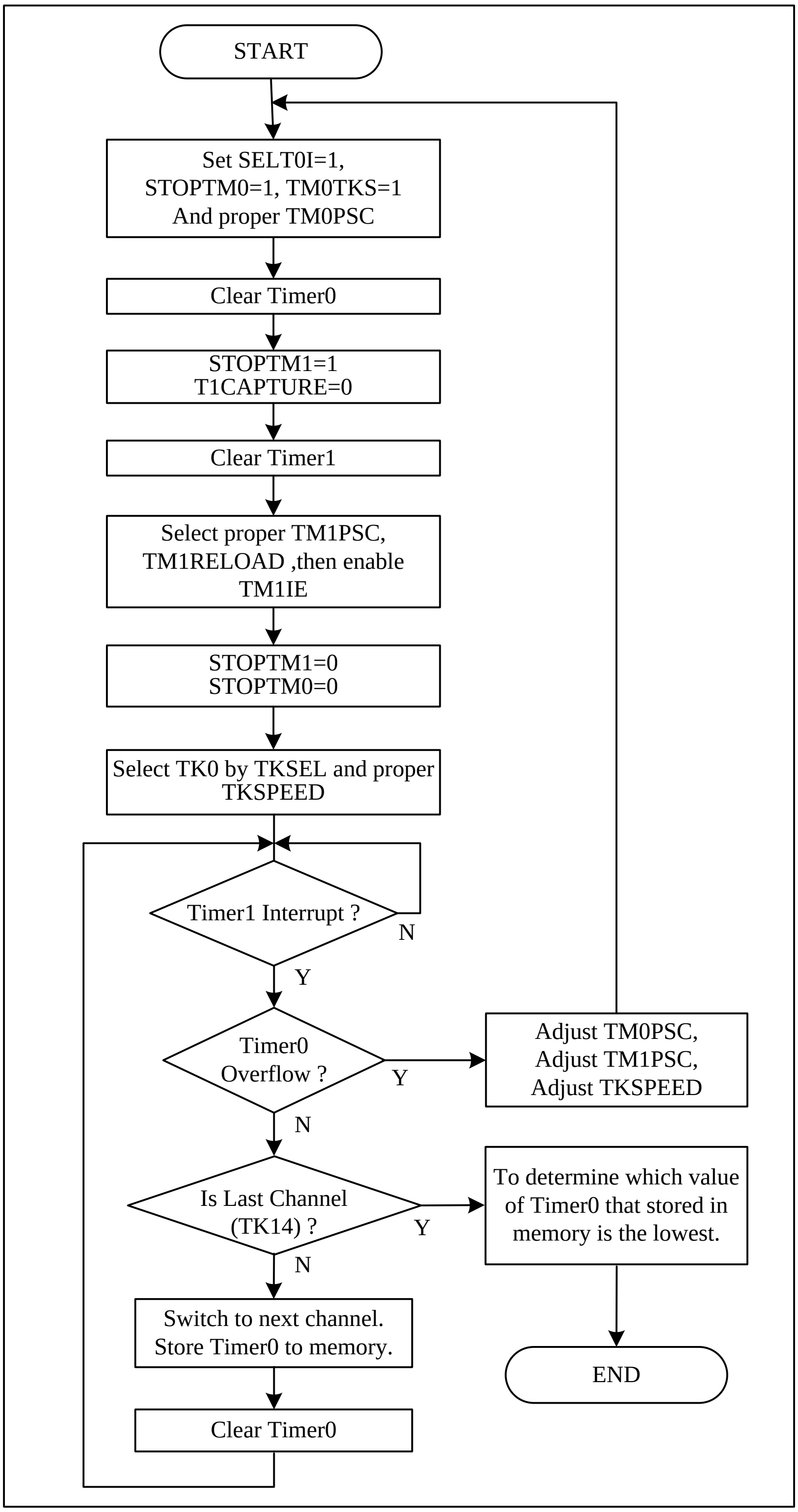


Between both TM0INT overflow depends on TKRC frequency, determine the amount of WKTINT to judge touched/ un-touched.



Between both WKTINT overflow depends on TKRC frequency, determine the amount of TM0INT to judge touched/un-touched.

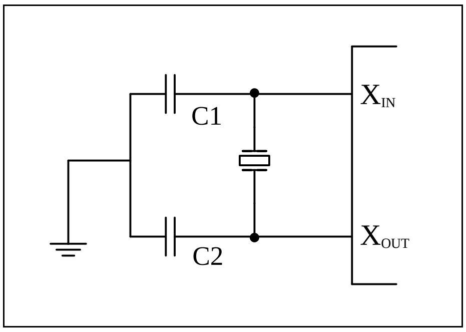
3.9.2 Touch Key using Timer0 and Timer1 timer


START
Set SELT0I=1, STOPTM0=1, TM0TKS=1 And proper TM0PSC
Clear Timer0
STOPTM1=1 T1CAPTURE=0
Clear Timer1
Select proper TM1PSC, TM1RELOAD ,then enable TM1IE
STOPTM1=0 STOPTM0=0
Select TK0 by TKSEL and proper TKSPEED
Timer1 Interrupt ?
N
Y
Timer0 Overflow ?
Y
N
Adjust TM0PSC, Adjust TM1PSC, Adjust TKSPEED
Is Last Channel (TK14) ?
Y
N
To determine which value of Timer0 that stored in memory is the lowest.
Switch to next channel. Store Timer0 to memory.
Clear Timer0
END

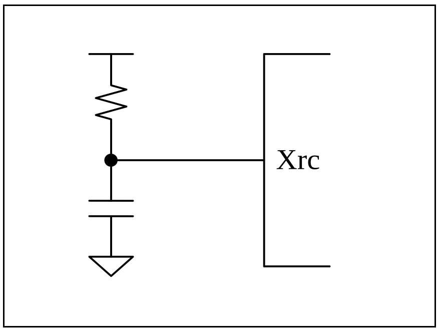
The recommended procedures

3.10 System Clock Oscillator

System clock can be operated in four different oscillation modes, which is selected by setting the CLKS in the SYSCFG register. In Slow/Fast Crystal (SXT/FXT) mode, a crystal or ceramic resonator is connected to the Xin and Xout pins to establish oscillation. In external RC (XRC) mode, the external resistor and capacitor determine the oscillation frequency. In the fast internal RC (FIRC) mode, the on-chip oscillator generates 16/8/4/2 MHz system clock, which controlled by register FIRCSEL[1:0] bits .



External Oscillator Circuit
(Crystal or Ceramic)

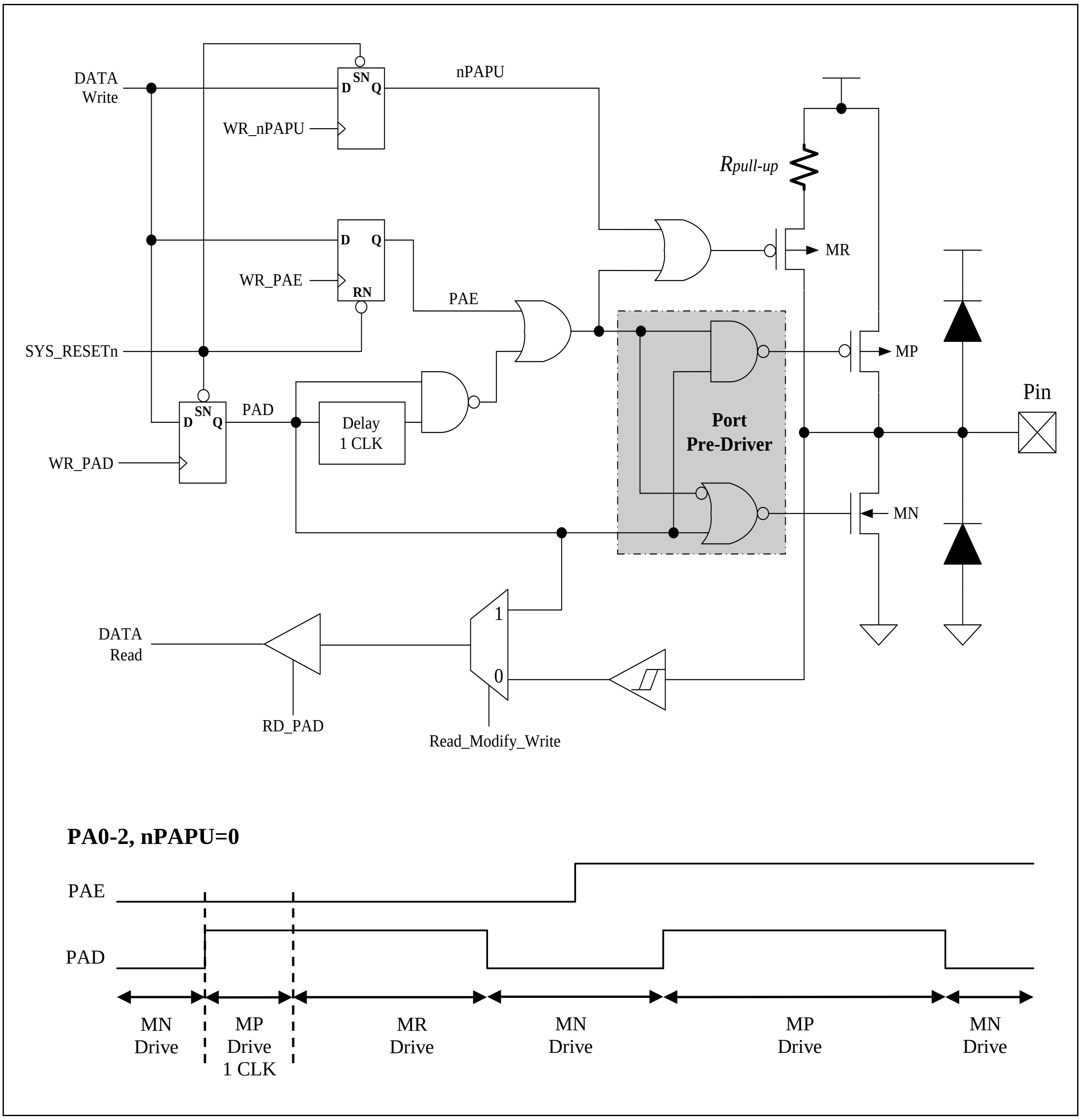


External RC Oscillator

4. I/O Port

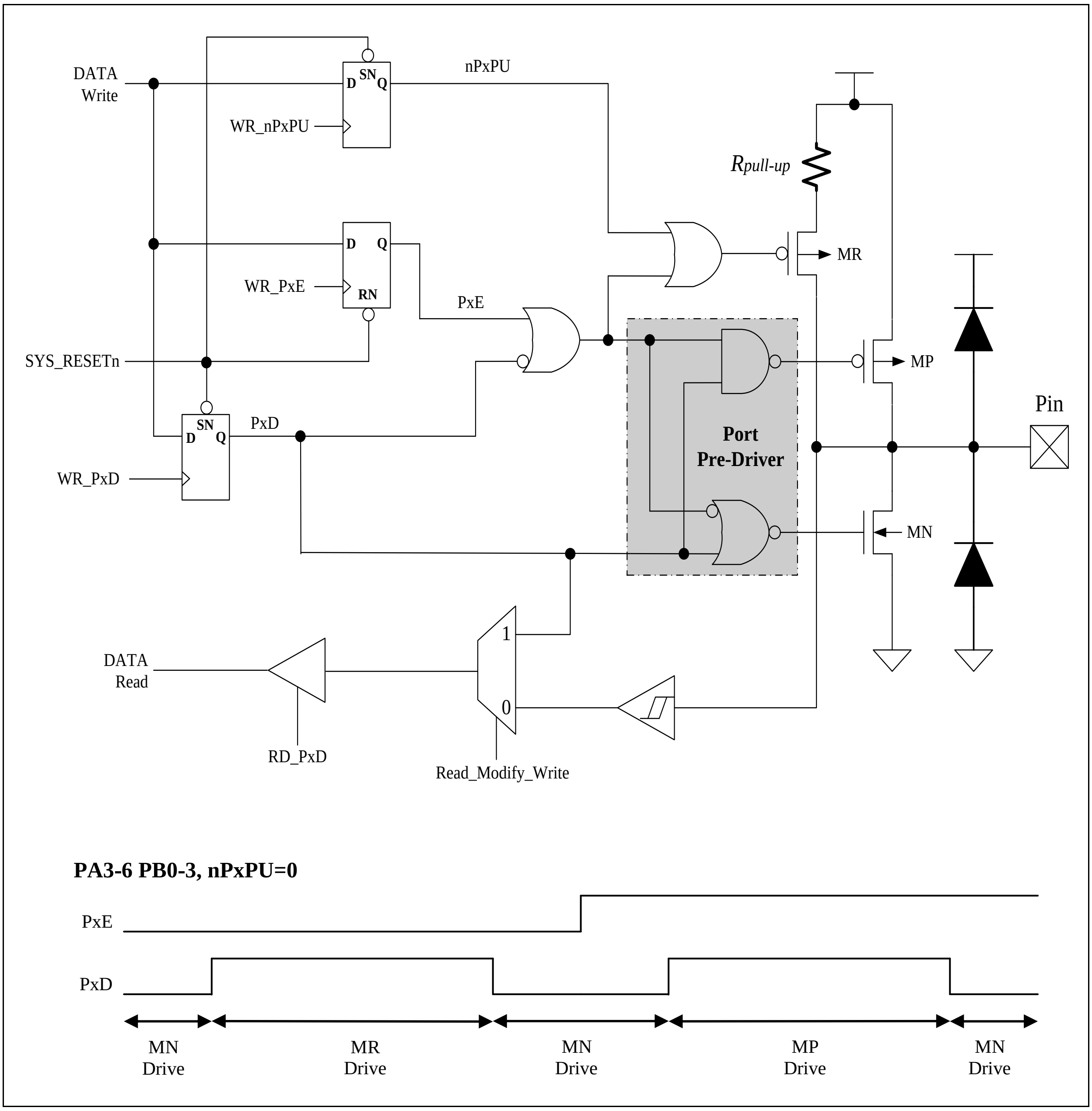
4.1 PA0-2

These pins can be used as Schmitt-trigger input, CMOS push-pull output or “pseudo-open-drain” output. The pull-up resistor is assignable to each pin by S/W setting. To use the pin in Schmitt-trigger input mode, S/W needs to set the PAE=0 and PAD=1. To use the pin in pseudo-open-drain mode, S/W sets the PAE=0.The benefit of pseudo-open-drain structure is that the output rise time can be much faster than pure open-drain structure. S/W sets PAE=1 to use the pin in CMOS push-pull output mode. Reading the pin data (PAD) has different meaning. In “Read-Modify-Write” instruction, CPU actually reads the output data register. In the others instructions, CPU reads the pin state. The so-called “Read-Modify-Write” instruction includes BSF, BCF and all instructions using F-Plane as destination.



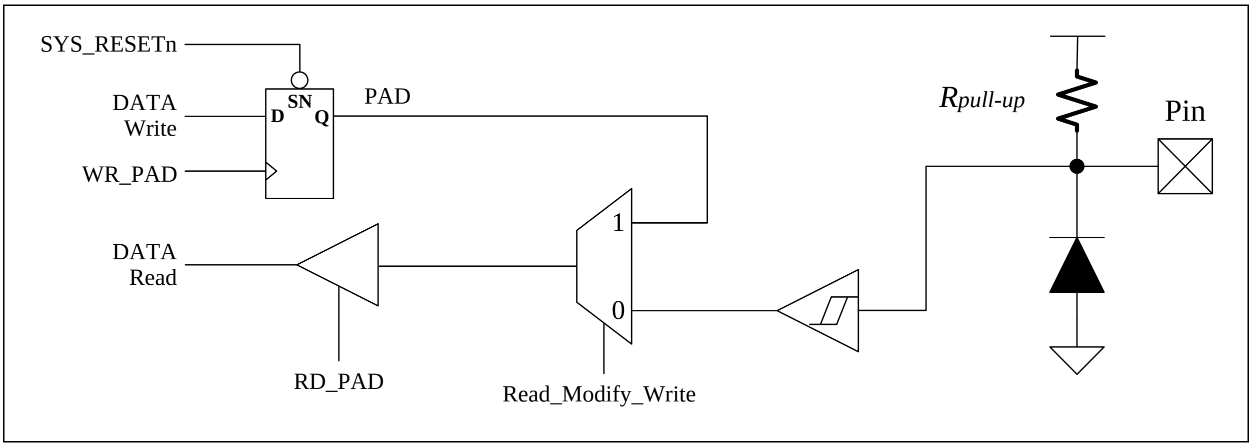
4.2 PA3-6, PB0-7, PD0-7, PE0-4

These pins are almost the same as PA0-2, except they do not support pseudo-open-drain mode. They can be used in pure open-drain mode, instead.



4.3 PA7

PA7 can be only used in Schmitt-trigger input mode. The pull-up resistor is always connected to this pin.



MEMORY MAP

F-Plane

Name	Adr	R/W	Rst	Description
INDF	00.7~0	R/W	-	Not a physical register, addressing INDF actually point to the register whose address is contained in the FSR register
TIMER0	01.7~0	R/W	0	Timer0 counting byte
PC	02.7~0	R/W	0	Programming Counter [7~0]
GBIT0	03.7	R/W	0	General purpose bit 0
GBIT1	03.6	R/W	0	General purpose bit 1
RAMBANK	03.5	R/W	0	RAM Bank Selection
TO	03.4	R	0	WDT timeout flag, cleared by PWRST, 'SLEEP' or 'CLRWDT' instruction
PD	03.3	R	0	Sleep mode flag, set by 'SLEEP', cleared by 'CLRWDT' instruction
ZFLAG	03.2	R/W	0	Zero flag
DCFLAG	03.1	R/W	0	Decimal Carry flag
CFLAG	03.0	R/W	0	Carry flag
GBIT2	04.7	R/W	0	General purpose bit 2
FSR	04.6~0	R/W	-	File Select Register, indirect address mode pointer
PAD7	05.7	R	-	PA7 pin state
PAD	05.6~0	R	-	Port A pin or "data register" state
		W	7F	Port A output data register
PBD	06.7~0	R	-	Port B pin or "data register" state
		W	FF	Port B output data register
PDD	07.7~0	R	-	Port D pin or "data register" state
		W	FF	Port D output data register
CMPIE	08.7	R/W	0	Comparator interrupt enable, 1=enable, 0=disable
TM2IE	08.6	R/W	0	Timer2 interrupt enable, 1=enable, 0=disable
TM1IE	08.5	R/W	0	Timer1 interrupt enable, 1=enable, 0=disable
TM0IE	08.4	R/W	0	Timer0 interrupt enable, 1=enable, 0=disable
WKTIE	08.3	R/W	0	Wakeup Timer interrupt enable, 1=enable, 0=disable Set 0 to clear & disable WKT timer
XINT2E	08.2	R/W	0	INT2 pin(PA2) interrupt enable, 1=enable, 0=disable
XINT1E	08.1	R/W	0	INT1 pin(PA1) interrupt enable, 1=enable, 0=disable
XINT0E	08.0	R/W	0	INT0 pin(PA0) interrupt enable, 1=enable, 0=disable
CMPIF	09.7	R	-	CMP int. event pending flag, set by CMP output result change
		W	0	write 0: clear this flag; write 1: no action
TM2IF	09.6	R	-	Timer2 interrupt event pending flag, set by H/W while Timer2 overflows
		W	0	write 0: clear this flag; write 1: no action
TM1IF	09.5	R	-	Timer1 interrupt event pending flag, set by H/W while Timer1 overflows
		W	0	write 0: clear this flag; write 1: no action
TM0IF	09.4	R	-	Timer0 interrupt event pending flag, set by H/W while Timer0 overflows
		W	0	write 0: clear this flag; write 1: no action

Name	Adr	R/W	Rst	Description
WKTIF	09.3	R	-	WKT interrupt event pending flag, set by H/W while WKT time out
		W	0	write 0: clear this flag; write 1: no action
XINT2IF	09.2	R	-	INT2 interrupt event pending flag, set by H/W at INT2 pin's falling edge
		W	0	write 0: clear this flag; write 1: no action
XINT1IF	09.1	R	-	INT1 interrupt event pending flag, set by H/W at INT1 pin's falling edge
		W	0	write 0: clear this flag; write 1: no action
XINT0IF	09.0	R	-	INT0 interrupt event pending flag, set by H/W at INT0 pin's f/r edge
		W	0	write 0: clear this flag; write 1: no action
TM1L	0a.7~0	R	-	Timer1 counting low byte
		W	0	Timer1 reload data low byte
TM1H	0b.7~0	R	-	Timer1 counting high byte
		W	0	Timer1 reload data high byte
PWMADTH	0c.7~0	R/W	0	PWMA duty 8-bit MSB
PWMADTL	0d.7~6	R/W	0	PWMA duty 2-bit LSB
-	0d.5~1	-	-	Reserved
CMPST	0d.0	R	-	Comparator compared result output
PWM0DUTY	0e.7~0	R/W	0	PWM0 duty 8-bit
-	0f.7	-	-	Reserved
SIRCSEL	0f.6~5	R/W	3	SIRC clock selection 00:128K 01:32K 10:8K 11:2 KHz
STPFCK	0f.4	R/W	0	1: Stop Fast-clock oscillating 0: Enable Fast-clock oscillate
SELSUB	0f.3	R/W	0	1: Force Slow-clock oscillate and Slow-clock as CPUCLK 0: Select Fast-clock as CPUCLK
SUBE	0f.2	R/W	0	If SELSUB=1, this SUBE bit is invalid, Slow-clock keep oscillating If SELSUB=0, Set 1 to enable Slow-clock oscillate, Clear 0 to stop Slow-clock oscillating
SUBTYP	0f.1~0	R/W	0	Slow-clock type 00:SXT 01:SIRC 10:XRC 11:TKRC
PED	10.4~0	R	-	Port E pin or “data register” state
		W	1F	Port E output data register
-	11.7~0	-	-	Reserved
-	12.7	-	-	Reserved
KICKE	12.6	R/W	1	Speedup SXT warmup, clear 0 to save power after SXT oscillate stable
CLRPWM0	12.5	R/W	1	PWM0 clear and hold
CLRTM2	12.4	R/W	1	Timer2 clear and hold when this bit is "1", default Timer2 is inactive
SETTM1	12.3	R/W	0	Timer1 set FFFFh and hold when this bit is "1"
CLRTM1	12.2	R/W	0	Clear Timer1 content and hold when this bit is "1"
STOPTM1	12.1	R/W	0	Stop Timer1 counting when this bit is "1"
STOPTM0	12.0	R/W	0	Stop Timer0 counting when this bit is "1"
SRAM	20~2f	R/W	-	RAM Common Area (16 bytes)
	30~7F	R/W	-	RAM BANK0 area (RAMBANK=0, 80 bytes)
	30~7F	R/W	-	RAM BANK1 area (RAMBANK=1, 80 bytes)

R-Plane

Name	Adr	R/W	Rst	Description
TM0PSC (R02h)				
CAPOLARITY	02.7	W	0	Timer0 Capture Mode Polarity 0:High level capture 1:Low level capture
T0CAPTURE	02.6	W	0	Timer0 Mode 0:Timer/Counter Mode (source from TM0PSC out or T0CKI or TKRC) 1:Capture Mode (source from CAPT pin)
T0IEDGE	02.5	W	0	Timer0 prescaler counting edge for T0CKI / TKRC 0: rising edge to increase Timer0 prescaler count 1: falling edge to increase Timer0 prescaler count
SELT0I	02.4	W	0	Timer0 prescaler clock source 0: clock source is Fcpuclk/2 (Instruction Cycle) 1: clock source is T0CKI pin (PE2 pin) or TKRC (see R-plane 0dh.6)
TM0PSC	02.3~0	W	0	Timer0 prescaler. Timer0 prescaler clock source divided by 0000: 1 0001: 2 0010: 4 0011: 8 0100: 16 0101: 32 0110: 64 0111: 128 1xxx: 256
PWRDOWN	03	W	-	write this register to enter Power-Down Mode (i.e. 'SLEEP' instruction)
CLRWDT	04	W	-	write this register to clear WDT timer
PAE	05.6~3	W	0	Each bit controls its corresponding pin, if the bit is 0: the pin is open-drain output or Schmitt-trigger input 1: the pin is CMOS push-pull output
	05.2~0	W	0	Each bit controls its corresponding pin, if the bit is 0: the pin is pseudo-open-drain output or Schmitt-trigger input 1: the pin is CMOS push-pull output
PBE	06.7~0	W	0	Each bit controls its corresponding pin, if the bit is 0: the pin is open-drain output or Schmitt-trigger input 1: the pin is CMOS push-pull output
PDE	07.7~0	W	0	Each bit controls its corresponding pin, if the bit is 0: the pin is open-drain output or Schmitt-trigger input 1: the pin is CMOS push-pull output
nPAPU	08.6~0	W	7F	Each bit controls its corresponding pin, if the bit is 0: the pin pull up resistor is enabled, except a. the pin's output data register (PAD) is 0 b. the pin's CMOS push-pull mode is chosen (PAE=1) c. the pin is working for FXT/SXT/XRC/PWM/TM0TGL output 1: the pin pull up resistor is disabled
nPBPU	09.7~0	W	FF	Each bit controls its corresponding pin, if the bit is 0: the pin pull up resistor is enabled, except a. the pin's output data register (PBD) is 0 b. the pin's CMOS push-pull mode is chosen (PBE=1) 1: the pin pull up resistor is disabled

Name	Adr	R/W	Rst	Description
nPDPU	0a.7~0	W	FF	Each bit controls its corresponding pin, if the bit is 0: the pin pull up resistor is enabled, except a. the pin's output data register (PDD) is 0 b. the pin's CMOS push-pull mode is chosen (PDE=1) c. pins are working for PWM/CMPO/TM1TGL/TCOUT output 1: the pin pull up resistor is disabled
WKTPSC (R0Bh)				
PWM0PSC	0b.7~6	W	0	PWM0 prescaler 0: Fcpuclk 1:Fcpuclk/2 2:Fcpuclk/4 3:Fcpuclk/8
PWMAE	0b.5	W	0	PWMA output to PA1/PD7 pin enable, see R-plane 17h.1
PWM0E	0b.4	W	0	PWM0 output to PA5/PD6 pin enable, see R-plane 17h.0
PWM0INV	0b.3	W	0	PWM0 negative output to pin 1:negative 0:positive
TM0OUT	0b.2	W	0	Timer0 match toggle out to PA6 pin enable
WKTPSC	0b.1~0	W	3	WKT period 00=0.9 ms, 01=1.8 ms, 10=25 ms, 11=100 ms (VDD=5V) 00=1.0 ms, 01=2.0 ms, 10=32 ms, 11=128 ms (VDD=3V)
TM1PSC (R0Ch)				
TCOPSC	0c7~6	W	0	CPUCLK prescaler 0:Fcpuclk/2 1:Fcpuclk/4 3:Fcpuclk/8 3:Fcpuclk/16
TCOE	0c.5	W	0	enable post-prescaler CPUCLK output to PD5 pin (TCOUT)
-	0c.4	-	-	Reserved
INT1EDGE	0c.3	W	0	0: INT1 pin falling edge to trigger interrupt event 1: INT1 pin rising edge to trigger interrupt event
TM1OE	0c.2	W	0	enable Timer1 overflow toggle output to PD0 pin (TM1TGL)
T1CAPTURE	0c.1	W	0	Timer1 Mode 0:Timer Mode (source form TM1PSC clock out) 1:Capture Mode (source from CAPT pin), measure CAPT pin period time between successive rising or falling edges
TM1PSC	0c.0	W	0	Timer1 prescaler. 1: Fcpuclk 0:Fcpuclk/2
TKCTL (R0Dh)				
WKTKRCS	0d.7	W	0	WKT timer clock select 0:SIRC 1:TKRC
TM0TKS	0d.6	W	0	Timer0 clock select 0:T0CKI pin or Fcpuclk/2 1:TKRC
TKSPEED	0d.5~4	W	0	Touch Key Oscillation Frequency Select, TKRC divider 0: Fastest Touch Key clock (div1) 1:div2 2:div8 3: Slowest Touch Key clock (div32)
TKSEL	0d.3~0	W	F	Touch Key Channel Select 0000: TK0 0001: TK1 ……… 1110: TK14 1111: No channel
TM2PSC (R0Eh)				
WDTPSC	0e.7~6	W	1	WDT time out. 0:112 ms 1:224 ms 2:896 ms 3:1800 ms
WDTSLPSTP	0e.5	W	0	1: WDT stop counting when IDLE/STOP mode 0:WDT always counting If WDTE=0, this WDTSLPSTP bit is invalid
TM2CLKS	0e.4	W	0	Timer2 clock source. 1:Fcpuclk/128 0:Slow-clock
FIRCSEL	0e.3~2	W	2	FIRC clock select. 3:16 MHz 2:4 MHz 1:8 MHz 0:2 MHz

Name	Adr	R/W	Rst	Description
TM2PSC	0e.1~0	W	0	Timer2 prescaler. Timer2 clock source divided by - 0:32768 1:16384 2:8192 3:128
TESTREG	0f.7~0	W	0	keep 0, don't write
CMPCTL (R10h)				
CMPEN	10.7	W	0	Comparator Enable
CMPEDGE	10.6	W	0	Comparator Interrupt Edge. 0:falling edge 1:rising edge
CMPOE	10.5	W	0	Comparator output to PD2 pin enable
CMPINNS	10.4	W	0	Comparator negative Input source select 1:IN1-(PD3) 0:IN0-(PD0)
CMPINPS	10.3~0	W	0	Comparator positive Input source select 0000:Vref0 0001:Vref1 0010:Vref2 …. 1110:Vref14 1111:IN+(PD1)
PWM0PROD	11.7~0	W	FF	PWM0 period
PB_IE	12.7~0	W	FF	Each bit controls its corresponding pin 0: disable I/O digital input to save power and multiplex in analog 1: enable I/O digital input
PD_IE	13.7~0	W	FF	Each bit controls its corresponding pin 0: disable I/O digital input to save power and multiplex in analog 1: enable I/O digital input
PBWKUP	14.7~0	W	0	1: PB7~0 wake up enable 0:disable
PEE	15.4~0	W	0	Each bit controls its corresponding pin, if the bit is 0: the pin is open-drain output or Schmitt-trigger input 1: the pin is CMOS push-pull output
nPEPU	16.4~0	W	FF	Each bit controls its corresponding pin, if the bit is 0: the pin pull up resistor is enable, except a. the pin's output data register (PED) is 0 b. the pin's CMOS push-pull mode is chosen (PEE=1) 1: the pin pull up resistor is disable
PWMA0S (R17h)				
-	17.7~2	-	-	Reserved
PWMA_PD7	17.1	W	0	PWMA output channel select 0:PA1 1:PD7
PWM0_PD6	17.0	W	0	PWM0 output channel select 0:PA5 1:PD6

INSTRUCTION SET

Each instruction is a 14-bit word divided into an Op Code, which specifies the instruction type, and one or more operands, which further specify the operation of the instruction. The instructions can be categorized as byte-oriented, bit-oriented and literal operations list in the following table.

For byte-oriented instructions, “f” or “r” represents the address designator and “d” represents the destination designator. The address designator is used to specify which address in Program memory is to be used by the instruction. The destination designator specifies where the result of the operation is to be placed. If “d” is “0”, the result is placed in the W register. If “d” is “1”, the result is placed in the address specified in the instruction.

For bit-oriented instructions, “b” represents a bit field designator, which selects the number of the bit affected by the operation, while “f” represents the address designator. For literal operations, “k” represents the literal or constant value.

Field / Legend	Description
f	F-Plane Register File Address
r	R-Plane Register File Address
b	Bit address
k	Literal. Constant data or label
d	Destination selection field. 0 : Working register 1 : Register file
W	Working Register
Z	Zero Flag
C	Carry Flag
DC	Decimal Carry Flag
PC	Program Counter
TOS	Top Of Stack
GIE	Global Interrupt Enable Flag (i-Flag)
[]	Option Field
( )	Contents
.	Bit Field
B	Before
A	After
←	Assign direction

Mnemonic	Op Code	Cycle	Flag Affect	Description
Byte-Oriented File Register Instruction				
ADDWF f,d	00 0111 dfff ffff	1	C,DC,Z	Add W and "f"
ANDWF f,d	00 0101 dfff ffff	1	Z	AND W with "f"
CLRF f	00 0001 1fff ffff	1	Z	Clear "f"
CLRW	00 0001 0100 0000	1	Z	Clear W
COMF f,d	00 1001 dfff ffff	1	Z	Complement "f"
DECF f,d	00 0011 dfff ffff	1	Z	Decrement "f"
DECFSZ f,d	00 1011 dfff ffff	1 or 2	-	Decrement "f", skip if zero
INCF f,d	00 1010 dfff ffff	1	Z	Increment "f"
INCFSZ f,d	00 1111 dfff ffff	1 or 2	-	Increment "f", skip if zero
IORWF f,d	00 0100 dfff ffff	1	Z	OR W with "f"
MOVFW f	00 1000 0fff ffff	1	-	Move "f" to W
MOVWF f	00 0000 1fff ffff	1	-	Move W to "f"
MOVWR r	00 0000 00rr rrrr	1	-	Move W to "r"
RLF f,d	00 1101 dfff ffff	1	C	Rotate left "f" through carry
RRF f,d	00 1100 dfff ffff	1	C	Rotate right "f" through carry
SUBWF f,d	00 0010 dfff ffff	1	C,DC,Z	Subtract W from "f"
SWAPF f,d	00 1110 dfff ffff	1	-	Swap nibbles in "f"
TESTZ f	00 1000 1fff ffff	1	Z	Test if "f" is zero
XORWF f,d	00 0110 dfff ffff	1	Z	XOR W with "f"
Bit-Oriented File Register Instruction				
BCF f,b	01 000b bbff ffff	1	-	Clear "b" bit of "f"
BSF f,b	01 001b bbff ffff	1	-	Set "b" bit of "f"
BTFSC f,b	01 010b bbff ffff	1 or 2	-	Test "b" bit of "f", skip if clear
BTFSS f,b	01 011b bbff ffff	1 or 2	-	Test "b" bit of "f", skip if set
Literal and Control Instruction				
ADDLW k	01 1100 kkkk kkkk	1	C,DC,Z	Add Literal "k" and W
ANDLW k	01 1011 kkkk kkkk	1	Z	AND Literal "k" with W
CALL k	10 kkkk kkkk kkkk	2	-	Call subroutine "k"
CLRWDT	00 0000 0000 0100	1	TO,PD	Clear Watch Dog Timer
GOTO k	11 kkkk kkkk kkkk	2	-	Jump to branch "k"
IORLW k	01 1010 kkkk kkkk	1	Z	OR Literal "k" with W
MOVLW k	01 1001 kkkk kkkk	1	-	Move Literal "k" to W
NOP	00 0000 0000 0000	1	-	No operation
RET	00 0000 0100 0000	2	-	Return from subroutine
RETI	00 0000 0110 0000	2	-	Return from interrupt
RETLW k	01 1000 kkkk kkkk	2	-	Return with Literal in W
SLEEP	00 0000 0000 0011	1	TO,PD	Go into standby mode, Clock oscillation stops
XORLW k	01 1111 kkkk kkkk	1	Z	XOR Literal "k" with W

ADDLW Add Literal "k" and W

Syntax	ADDLW k	
Operands	k : 00h ~ FFh	
Operation	(W) ← (W) + k	
Status Affected	C, DC, Z	
OP-Code	01 1100 kkkk kkkk	
Description	The contents of the W register are added to the eight-bit literal 'k' and the result is placed in the W register.	
Cycle	1	
Example	ADDLW 0x15	B : W = 0x10 A : W = 0x25

ADDWF Add W and "f"

Syntax	ADDWF f [,d]	
Operands	f : 00h ~ 7Fh, d : 0, 1	
Operation	(destination) ← (W) + (f)	
Status Affected	C, DC, Z	
OP-Code	00 0111 dfff ffff	
Description	Add the contents of the W register with register 'f'. If 'd' is 0, the result is stored in the W register. If 'd' is 1, the result is stored back in register 'f'.	
Cycle	1	
Example	ADDWF FSR, 0	B : W = 0x17, FSR = 0xC2 A : W = 0xD9, FSR = 0xC2

ANDLW Logical AND Literal "k" with W

Syntax	ANDLW k	
Operands	k : 00h ~ FFh	
Operation	(W) ← (W) AND k	
Status Affected	Z	
OP-Code	01 1011 kkkk kkkk	
Description	The contents of W register are AND'ed with the eight-bit literal 'k'. The result is placed in the W register.	
Cycle	1	
Example	ANDLW 0x5F	B : W = 0xA3 A : W = 0x03

ANDWF AND W with "f"

Syntax	ANDWF f [,d]	
Operands	f : 00h ~ 7Fh, d : 0, 1	
Operation	(destination) ← (W) AND (f)	
Status Affected	Z	
OP-Code	00 0101 dfff ffff	
Description	AND the W register with register 'f'. If 'd' is 0, the result is stored in the W register. If 'd' is 1, the result is stored back in register 'f'.	
Cycle	1	
Example	ANDWF FSR, 1	B : W = 0x17, FSR = 0xC2 A : W = 0x17, FSR = 0x02

BCF — Clear "b" bit of "f"

Syntax	BCF f [,b]	
Operands	f : 00h ~ 3Fh, b : 0 ~ 7	
Operation	(f.b) ← 0	
Status Affected	-	
OP-Code	01 000b bbff ffff	
Description	Bit 'b' in register 'f' is cleared.	
Cycle	1	
Example	BCF FLAG_REG, 7	B : FLAG_REG = 0xC7
		A : FLAG_REG = 0x47

BSF — Set "b" bit of "f"

Syntax	BSF f [,b]	
Operands	f : 00h ~ 3Fh, b : 0 ~ 7	
Operation	(f.b) ← 1	
Status Affected	-	
OP-Code	01 001b bbff ffff	
Description	Bit 'b' in register 'f' is set.	
Cycle	1	
Example	BSF FLAG_REG, 7	B : FLAG_REG = 0x0A
		A : FLAG_REG = 0x8A

BTFSC — Test "b" bit of "f", skip if clear(0)

Syntax	BTFSC f [,b]	
Operands	f : 00h ~ 3Fh, b : 0 ~ 7	
Operation	Skip next instruction if (f.b) = 0	
Status Affected	-	
OP-Code	01 010b bbff ffff	
Description	If bit 'b' in register 'f' is 1, then the next instruction is executed. If bit 'b' in register 'f' is 0, then the next instruction is discarded, and a NOP is executed instead, making this a 2nd cycle instruction.	
Cycle	1 or 2	
Example	LABEL1 BTFSC FLAG, 1	B : PC = LABEL1
	TRUE GOTO SUB1	A : if FLAG.1 = 0, PC = FALSE
	FALSE ...	if FLAG.1 = 1, PC = TRUE

BTFSS — Test "b" bit of "f", skip if set(1)

Syntax	BTFSS f [,b]	
Operands	f : 00h ~ 3Fh, b : 0 ~ 7	
Operation	Skip next instruction if (f.b) = 1	
Status Affected	-	
OP-Code	01 011b bbff ffff	
Description	If bit 'b' in register 'f' is 0, then the next instruction is executed. If bit 'b' in register 'f' is 1, then the next instruction is discarded, and a NOP is executed instead, making this a 2nd cycle instruction.	
Cycle	1 or 2	
Example	LABEL1 BTFSS FLAG, 1	B : PC = LABEL1
	TRUE GOTO SUB1	A : if FLAG.1 = 0, PC = TRUE
	FALSE ...	if FLAG.1 = 1, PC = FALSE

CALL	**Call subroutine "k"**	
Syntax	CALL k	
Operands	k : 000h ~ FFFh	
Operation	Operation: TOS ← (PC) + 1, PC.11~0 ← k	
Status Affected	-	
OP-Code	10 kkkk kkkk kkkk	
Description	Call Subroutine. First, return address (PC+1) is pushed onto the stack. The 12-bit immediate address is loaded into PC bits <11:0>. CALL is a two-cycle instruction.	
Cycle	2	
Example	LABEL1 CALL SUB1	B : PC = LABEL1 A : PC = SUB1, TOS = LABEL1 + 1

CLRF	**Clear "f"**	
Syntax	CLRF f	
Operands	f : 00h ~ 7Fh	
Operation	(f) ← 00h, Z ← 1	
Status Affected	Z	
OP-Code	00 0001 1fff ffff	
Description	The contents of register 'f' are cleared and the Z bit is set.	
Cycle	1	
Example	CLRF FLAG_REG	B : FLAG_REG = 0x5A A : FLAG_REG = 0x00, Z = 1

CLRW	**Clear W**	
Syntax	CLRW	
Operands	-	
Operation	(W) ← 00h, Z ← 1	
Status Affected	Z	
OP-Code	00 0001 0100 0000	
Description	W register is cleared and Z bit is set.	
Cycle	1	
Example	CLRW	B : W = 0x5A A : W = 0x00, Z = 1

CLRWDT	**Clear Watchdog Timer**	
Syntax	CLRWDT	
Operands	-	
Operation	WDT Timer ← 00h	
Status Affected	TO, PD	
OP-Code	00 0000 0000 0100	
Description	CLRWDT instruction clears the Watchdog Timer	
Cycle	1	
Example	CLRWDT	B : WDT counter = ? A : WDT counter = 0x00

COMF	Complement "f"	
Syntax	COMF f [,d]	
Operands	f : 00h ~ 7Fh, d : 0, 1	
Operation	(destination) ← $(\overline{f})$	
Status Affected	Z	
OP-Code	00 1001 dfff ffff	
Description	The contents of register 'f' are complemented. If 'd' is 0, the result is stored in W. If 'd' is 1, the result is stored back in register 'f'.	
Cycle	1	
Example	COMF REG1, 0	B : REG1 = 0x13 A : REG1 = 0x13, W = 0xEC

DECF	Decrement "f"	
Syntax	DECF f [,d]	
Operands	f : 00h ~ 7Fh, d : 0, 1	
Operation	(destination) ← (f) - 1	
Status Affected	Z	
OP-Code	00 0011 dfff ffff	
Description	Decrement register 'f'. If 'd' is 0, the result is stored in the W register. If 'd' is 1, the result is stored back in register 'f'.	
Cycle	1	
Example	DECF CNT, 1	B : CNT = 0x01, Z = 0 A : CNT = 0x00, Z = 1

DECFSZ	Decrement "f", Skip if 0	
Syntax	DECFSZ f [,d]	
Operands	f : 00h ~ 7Fh, d : 0, 1	
Operation	(destination) ← (f) - 1, skip next instruction if result is 0	
Status Affected	-	
OP-Code	00 1011 dfff ffff	
Description	The contents of register 'f' are decremented. If 'd' is 0, the result is placed in the W register. If 'd' is 1, the result is placed back in register 'f'. If the result is 1, the next instruction is executed. If the result is 0, then a NOP is executed instead, making it a 2 cycle instruction.	
Cycle	1 or 2	

Example	LABEL1 DECFSZ CNT, 1
	GOTO LOOP
	CONTINUE
		B : PC = LABEL1
		A : CNT = CNT – 1
		if CNT = 0, PC = CONTINUE
		if CNT ≠ 0, PC = LABEL1 + 1

GOTO	Unconditional Branch	
Syntax	GOTO k	
Operands	k : 000h ~ FFFh	
Operation	PC.11~0 ← k	
Status Affected	-	
OP-Code	11 kkkk kkkk kkkk	
Description	GOTO is an unconditional branch. The 12-bit immediate value is loaded into PC bits <11:0>. GOTO is a two-cycle instruction.	
Cycle	2	
Example	LABEL1 GOTO SUB1	B : PC = LABEL1 A : PC = SUB1

INCF Increment "f"

Syntax	INCF f [,d]	
Operands	f : 00h ~ 7Fh	
Operation	(destination) ← (f) + 1	
Status Affected	Z	
OP-Code	00 1010 dfff ffff	
Description	The contents of register 'f' are incremented. If 'd' is 0, the result is placed in the W register. If 'd' is 1, the result is placed back in register 'f'.	
Cycle	1	
Example	INCF CNT, 1	B : CNT = 0xFF, Z = 0 A : CNT = 0x00, Z = 1

INCFSZ Increment "f", Skip if 0

Syntax	INCFSZ f [,d]	
Operands	f : 00h ~ 7Fh, d : 0, 1	
Operation	(destination) ← (f) + 1, skip next instruction if result is 0	
Status Affected	-	
OP-Code	00 1111 dfff ffff	
Description	The contents of register 'f' are incremented. If 'd' is 0, the result is placed in the W register. If 'd' is 1, the result is placed back in register 'f'. If the result is 1, the next instruction is executed. If the result is 0, a NOP is executed instead, making it a 2 cycle instruction.	
Cycle	1 or 2	

Example	LABEL1 INCFSZ CNT, 1
	GOTO LOOP
	CONTINUE
		B : PC = LABEL1
		A : CNT = CNT + 1
		if CNT = 0, PC = CONTINUE
		if CNT ≠ 0, PC = LABEL1 + 1

IORLW Inclusive OR Literal with W

Syntax	IORLW k	
Operands	k : 00h ~ FFh	
Operation	(W) ← (W) OR k	
Status Affected	Z	
OP-Code	01 1010 kkkk kkkk	
Description	The contents of the W register are OR'ed with the eight-bit literal 'k'. The result is placed in the W register.	
Cycle	1	
Example	IORLW 0x35	B : W = 0x9A A : W = 0xBF, Z = 0

IORWF Inclusive OR W with "f"

Syntax	IORWF f [,d]	
Operands	f : 00h ~ 7Fh, d : 0, 1	
Operation	(destination) ← (W) OR k	
Status Affected	Z	
OP-Code	00 0100 dfff ffff	
Description	Inclusive OR the W register with register 'f'. If 'd' is 0, the result is placed in the W register. If 'd' is 1, the result is placed back in register 'f'.	
Cycle	1	
Example	IORWF RESULT, 0	B : RESULT = 0x13, W = 0x91 A : RESULT = 0x13, W = 0x93, Z = 0

MOVFW	Move "f" to W	
Syntax	MOVFW f	
Operands	f : 00h ~ 7Fh	
Operation	(W) ← (f)	
Status Affected	-	
OP-Code	00 1000 0fff ffff	
Description	The contents of register 'f' are moved to W register.	
Cycle	1	
Example	MOVFW FSR	B : FSR = 0xC2, W = ? A : FSR = 0xC2, W = 0xC2

MOVLW	Move Literal to W	
Syntax	MOVLW k	
Operands	k : 00h ~ FFh	
Operation	(W) ← k	
Status Affected	-	
OP-Code	01 1001 kkkk kkkk	
Description	The eight-bit literal 'k' is loaded into W register. The don’t cares will assemble as 0’s.	
Cycle	1	
Example	MOVLW 0x5A	B : W = ? A : W = 0x5A

MOVWF	Move W to "f"	
Syntax	MOVWF f	
Operands	f : 00h ~ 7Fh	
Operation	(f) ← (W)	
Status Affected	-	
OP-Code	00 0000 1fff ffff	
Description	Move data from W register to register 'f'.	
Cycle	1	
Example	MOVWF REG1	B : REG1 = 0xFF, W = 0x4F A : REG1 = 0x4F, W = 0x4F

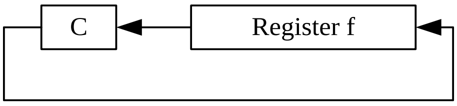
MOVWR	Move W to "r"	
Syntax	MOVWR r	
Operands	r : 00h ~ 3Fh	
Operation	(r) ← (W)	
Status Affected	-	
OP-Code	00 0000 00rr rrrr	
Description	Move data from W register to register 'r'.	
Cycle	1	
Example	MOVWR REG1	B : REG1 = 0xFF, W = 0x4F A : REG1 = 0x4F, W = 0x4F

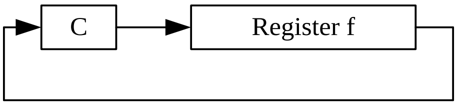
NOP	**No Operation**	
Syntax	NOP	
Operands	-	
Operation	No Operation	
Status Affected	-	
OP-Code	00 0000 0000 0000	
Description	No Operation	
Cycle	1	
Example	NOP	-

RET	**Return from Subroutine**	
Syntax	RET	
Operands	-	
Operation	PC ← TOS	
Status Affected	-	
OP-Code	00 0000 0100 0000	
Description	Return from subroutine. The stack is POPed and the top of the stack (TOS) is loaded into the program counter. This is a two-cycle instruction.	
Cycle	2	
Example	RET	A : PC = TOS

RETI	**Return from Interrupt**	
Syntax	RETI	
Operands	-	
Operation	PC ← TOS, GIE ← 1	
Status Affected	-	
OP-Code	00 0000 0110 0000	
Description	Return from Interrupt. Stack is POPed and Top-of-Stack (TOS) is loaded in to the PC. Interrupts are enabled. This is a two-cycle instruction.	
Cycle	2	
Example	RETI	A : PC = TOS, GIE = 1

RETLW	**Return with Literal in W**	
Syntax	RETLW k	
Operands	k : 00h ~ FFh	
Operation	PC ← TOS, (W) ← k	
Status Affected	-	
OP-Code	01 1000 kkkk kkkk	
Description	The W register is loaded with the eight-bit literal 'k'. The program counter is loaded from the top of the stack (the return address). This is a two-cycle instruction.	
Cycle	2	
Example	CALL TABLE	B : W = 0x07
	:	A : W = value of k8
	TABLE ADDWF PCL, 1	
	RETLW k1	
	RETLW k2	
	:	
	RETLW kn	

RLF	Rotate Left "f" through Carry
Syntax	RLF f [,d]
Operands	f : 00h ~ 7Fh, d : 0, 1
Operation	
Status Affected	C
OP-Code	00 1101 dfff ffff
Description	The contents of register 'f' are rotated one bit to the left through the Carry Flag. If 'd' is 0, the result is placed in the W register. If 'd' is 1, the result is stored back in register 'f'.
Cycle	1
Example	RLF REG1, 0 B : REG1 = 1110 0110, C = 0 A : REG1 = 1110 0110 W = 1100 1100, C = 1

RRF	Rotate Right "f" through Carry
Syntax	RRF f [,d]
Operands	f : 00h ~ 7Fh, d : 0, 1
Operation	
Status Affected	C
OP-Code	00 1100 dfff ffff
Description	The contents of register 'f' are rotated one bit to the right through the Carry Flag. If 'd' is 0, the result is placed in the W register. If 'd' is 1, the result is placed back in register 'f'.
Cycle	1
Example	RRF REG1, 0 B : REG1 = 1110 0110, C = 0 A : REG1 = 1110 0110 W = 0111 0011, C = 0

SLEEP	Go into standby mode, Clock oscillation stops
Syntax	SLEEP
Operands	-
Operation	-
Status Affected	TO, PD
OP-Code	00 0000 0000 0011
Description	Go into SLEEP mode with the oscillator stopped.
Cycle	1
Example	SLEEP -

SUBWF Subtract W from "f"

Syntax	SUBWF f [,d]	
Operands	f : 00h ~ 7Fh, d : 0, 1	
Operation	(destination) ← (f) – (W)	
Status Affected	C, DC, Z	
OP-Code	00 0010 dfff ffff	
Description	Subtract (2’s complement method) W register from register 'f'. If 'd' is 0, the result is stored in the W register. If 'd' is 1, the result is stored back in register 'f'.	
Cycle	1	
Example	SUBWF REG1, 1	B : REG1 = 0x03, W = 0x02, C = ?, Z = ? A : REG1 = 0x01, W = 0x02, C = 1, Z = 0
	SUBWF REG1, 1	B : REG1 = 0x02, W = 0x02, C = ?, Z = ? A : REG1 = 0x00, W = 0x02, C = 1, Z = 1
	SUBWF REG1, 1	B : REG1 = 0x01, W = 0x02, C = ?, Z = ? A : REG1 = 0xFF, W = 0x02, C = 0, Z = 0

SWAPF Swap Nibbles in "f"

Syntax	SWAPF f [,d]	
Operands	f : 00h ~ 7Fh, d : 0, 1	
Operation	(destination,7~4) ← (f.3~0), (destination.3~0) ← (f.7~4)	
Status Affected	-	
OP-Code	00 1110 dfff ffff	
Description	The upper and lower nibbles of register 'f' are exchanged. If 'd' is 0, the result is placed in W register. If 'd' is 1, the result is placed in register 'f'.	
Cycle	1	
Example	SWAPF REG1, 0	B : REG1 = 0xA5 A : REG1 = 0xA5, W = 0x5A

TESTZ Test if "f" is zero

Syntax	TESTZ f	
Operands	f : 00h ~ 7Fh	
Operation	Set Z flag if (f) is 0	
Status Affected	Z	
OP-Code	00 1000 1fff ffff	
Description	If the content of register 'f' is 0, Zero flag is set to 1.	
Cycle	1	
Example	TESTZ REG1	B : REG1 = 0, Z = ? A : REG1 = 0, Z = 1

XORLW Exclusive OR Literal with W

Syntax	XORLW k	
Operands	k : 00h ~ FFh	
Operation	(W) ← (W) XOR k	
Status Affected	Z	
OP-Code	01 1111 kkkk kkkk	
Description	The contents of the W register are XOR'ed with the eight-bit literal 'k'. The result is placed in the W register.	
Cycle	1	
Example	XORLW 0xAF	B : W = 0xB5 A : W = 0x1A

XORWF Exclusive OR W with "f"

Syntax	XORWF f [,d]	
Operands	f : 00h ~ 7Fh, d : 0, 1	
Operation	(destination) ← (W) XOR (f)	
Status Affected	Z	
OP-Code	00 0110 dfff ffff	
Description	Exclusive OR the contents of the W register with register 'f'. If 'd' is 0, the result is stored in the W register. If 'd' is 1, the result is stored back in register 'f'.	
Cycle	1	
Example	XORWF REG, 1	B : REG = 0xAF, W = 0xB5 A : REG = 0x1A, W = 0xB5

ELECTRICAL CHARACTERISTICS

1. Absolute Maximum Ratings (T_A = 25℃)

Parameter	Rating	Unit
Supply voltage	V_{SS} - 0.3 to V_{SS} + 6.5	V
Input voltage	V_{SS} - 0.3 to V_{DD} + 0.3	
Output voltage	V_{SS} - 0.3 to V_{DD} + 0.3	
Output current high per 1 PIN	-25	mA
Output current high per all PIN	-80	
Output current low per 1 PIN	+30	
Output current low per all PIN	+150	
Maximum Operating Voltage	5.5	V
Operating temperature	-40 to +85	°C
Storage temperature	-65 to +150	

2. DC Characteristics (T_A = 25°C, V_{DD} = 2.0 V to 5.5 V)

Parameter	Sym	Conditions		Min	Typ	Max	Unit
Operating Voltage	V_{DD}	FAST mode, 25°C, F_{OSC} = 24 MHz		4.0	–	5.5	V
		FAST mode, 25°C, F_{OSC} = 16 MHz		3.1	–	5.5	
		FAST mode, 25°C, F_{OSC} = 12MHz		2.1	–	5.5	
		FAST mode, 25°C, F_{OSC} = 8 MHz		1.8	–	5.5	
		FAST mode, 25°C, F_{OSC} = 4 MHz		1.7	–	5.5	
		SLOW mode, 25°C, F_{OSC} < 3 MHz		1.5	–	5.5	
Input High Voltage	V_{IH}	All Input, except PA7	V_{DD} = 5 V	0.6V_{DD}	–	–	V
			V_{DD} = 3 V	0.6V_{DD}	–	–	V
		PA7	V_{DD} = 5 V	0.7V_{DD}	–	–	V
			V_{DD} = 3 V	0.7V_{DD}	–	–	V
Input Low Voltage	V_{IL}	All Input, except PA7	V_{DD} = 5 V	–	–	0.2V_{DD}	V
			V_{DD} = 3 V	–	–	0.2V_{DD}	V
		PA7	V_{DD} = 5 V	–	–	0.2V_{DD}	V
			V_{DD} = 3 V	–	–	0.2V_{DD}	V
Output High Voltage	V_{OH}	All Output	V_{DD} = 5 V, I_{OH} = 8 mA	4.4	–	–	V
			V_{DD} = 3 V, I_{OH} = 4 mA	2.6	–	–	V
Output Low Voltage	V_{OL}	All Output	V_{DD} = 5 V, I_{OL} = 20 mA	–	–	0.5	V
			V_{DD} = 3 V, I_{OL} = 10 mA	–	–	0.3	V
Input Leakage Current (pin high)	I_{ILH}	All Input	V_{IN} = V_{DD}	–	–	1	μA
Input Leakage Current (pin low)	I_{ILL}	All Input	V_{IN} = 0 V	–	–	-1	μA
Supply Current	I_{DD}	FAST mode, LVR enable, WDT enable No Load	V_{DD} = 5 V, FXT = 16 MHz	–	7	–	mA
			V_{DD} = 3 V, FXT = 16 MHz	–	3	–	
			V_{DD} = 5 V, FXT = 8 MHz	–	4	–	
			V_{DD} = 3 V, FXT = 8 MHz	–	2	–	
			V_{DD} = 5 V, FXT = 4 MHz	–	2.5	–	
			V_{DD} = 3 V, FXT = 4 MHz	–	1	–	
			V_{DD} = 5 V, FIRC = 16 MHz	–	6	–	
			V_{DD} = 5 V, FIRC = 8 MHz	–	3.2	–	
			V_{DD} = 3 V, FIRC = 8 MHz	–	1.5	–	
			V_{DD} = 5 V, FIRC = 4 MHz	–	2	–	
			V_{DD} = 3 V, FIRC = 4 MHz	–	1	–	

Parameter	Sym	Conditions		Min	Typ	Max	Unit
Supply Current	I_{DD}	SLOW mode, LVR enable, WKTPSC=11	V_{DD} = 5 V, SXT = 32 KHz	–	60	–	μA
			V_{DD} = 3 V, SXT = 32 KHz	–	10	–	
			V_{DD} = 5 V, SIRC=2KHz	–	20	–	
			V_{DD} = 3 V, SIRC=2KHz	–	6	–	
		IDLE mode, LVR enable	V_{DD} = 5 V, SXT=32 KHz	–	35	–	
			V_{DD} = 3 V, SXT=32 KHz TM2PSC=2’b00	–	3	–	
			V_{DD} = 5 V, SIRC=32KHz	–	34	–	
			V_{DD} = 3 V, SIRC=32KHz	–	3	–	
		IDLE mode, LVR disable	V_{DD} = 5 V, SXT = 32 KHz	–	34	–	
			V_{DD} = 3 V, SXT=32 KHz TM2PSC=2’b00	–	2	–	
			V_{DD} = 5 V, SIRC=32KHz	–	33	–	
			V_{DD} = 3 V, SIRC=32KHz	–	2	–	
		STOP mode, LVR enable	V_{DD} = 5 V	–	1	2	
			V_{DD} = 3 V	–	0.5	1	
		STOP mode, LVR disable	V_{DD} = 5 V	–	–	0.1	
			V_{DD} = 3 V	–	–	0.1	
Pull-Up Resistor	R_P	V_{IN} = 0 V Ports A/B/D/E	V_{DD} = 5 V V_{DD} = 3 V	–	60 120	–	KΩ
		V_{IN} = 0 V PA7	V_{DD} = 5 V V_{DD} = 3 V	–	55 55	–	KΩ

3. Clock Timing (T_A = -40℃ to +85℃)

Parameter	Condition			Min	Typ	Max	Unit
External RC Frequency	V_{DD} = 3V	R = 4.7K	C = 20 pF	–	3.3	–	MHz
		R = 10K	C = 100 pF	–	0.8	–	
		R = 100K	C = 300 pF	–	0.04	–	
	V_{DD} = 5V	R = 4.7K	C = 20 pF	–	3.6	–	
		R = 10K	C = 100 pF	–	0.7	–	
		R = 100K	C = 300 pF	–	0.03	–	
FIRC Frequency	25℃, V_{DD} = 3 ~ 5.5 V (±3%)			7.75	8	8.25	
	25℃, V_{DD} = 2.6 ~ 3 V (±5%)			7.6	8	8.4	
	-40℃ ~ 85℃, V_{DD} = 2.6 ~ 5.5 V			7.5	8	8.5	

4. Reset Timing Characteristics (T_A = 25℃)

Parameter	Conditions	Min	Typ	Max	Unit
RESET Input Low width	Input V_{DD} = 5 V ± 10%	3	–	–	μs
WDT time	V_{DD} = 5 V, WDTPSC = 00	–	100	–	ms
	V_{DD} = 3 V, WDTPSC = 00	–	130	–	ms
WKT time	V_{DD} = 5 V, WKTPSC = 00	–	0.9	–	ms
	V_{DD} = 3 V, WKTPSC = 00	–	1.0	–	ms
CPU start up time	V_{DD} = 5 V	–	14	–	ms

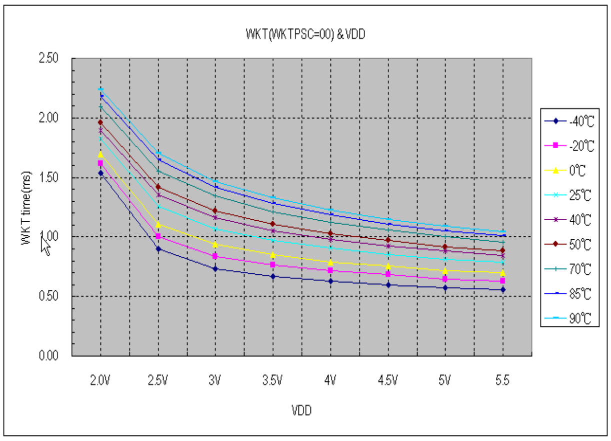
5. LVR Circuit Characteristics (T_A = 25℃)

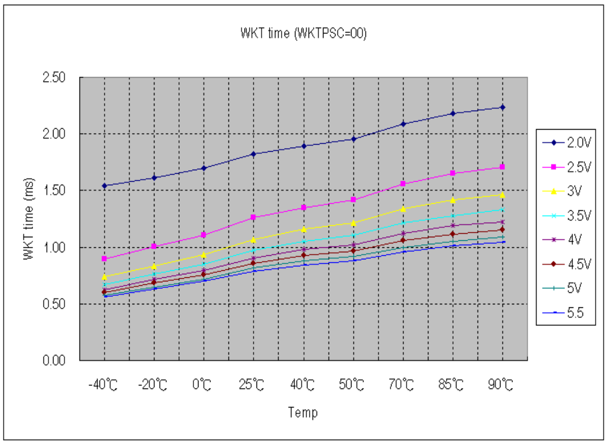
Parameter	Symbol	Min	Typ	Max	Unit
LVR Reference Voltage	V_{LVR}	–	1.5	–	V
		–	2.3	–	
		–	3.2	–	
LVR Hysteresis Voltage	V_{HYST}	–	±0.1	–	V
Low Voltage Detection time	t_{LVR}	100	–	–	μs

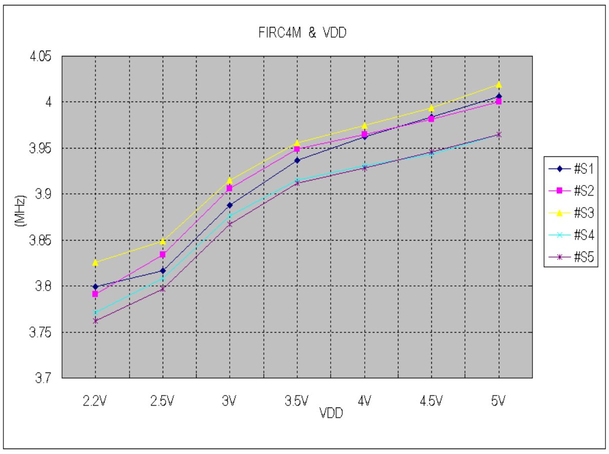
6. Comparator Characteristics (T_A = 25℃)

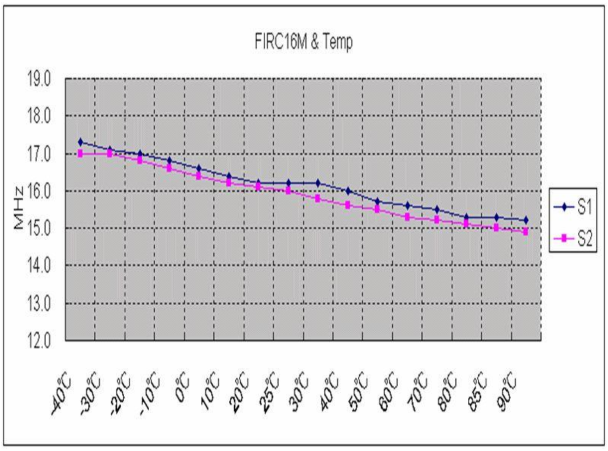
Parameter	Symbol	Conditions	Min	Typ	Max	Unit
CMP Operating Current	I_{CMP}	V_{DD} = 5V	–	40		uA
CMP Input Offset	V_{OFFSET}	Vin=VSS	-50	–	+50	mV
CMP Common Volt Range	V_{CCVR}	V_{DD} = 5V	V_{SS} - 0.3	–	V_{DD} - 0.3	V
Internal Vref Voltage	V_{CIVREF}	CMPINPS=0~14	V_{SS} + 0.1		V_{DD} - 0.6	V

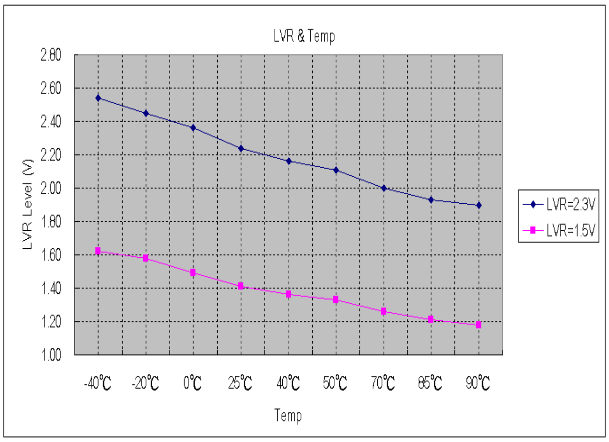
7. Characteristic Graphs









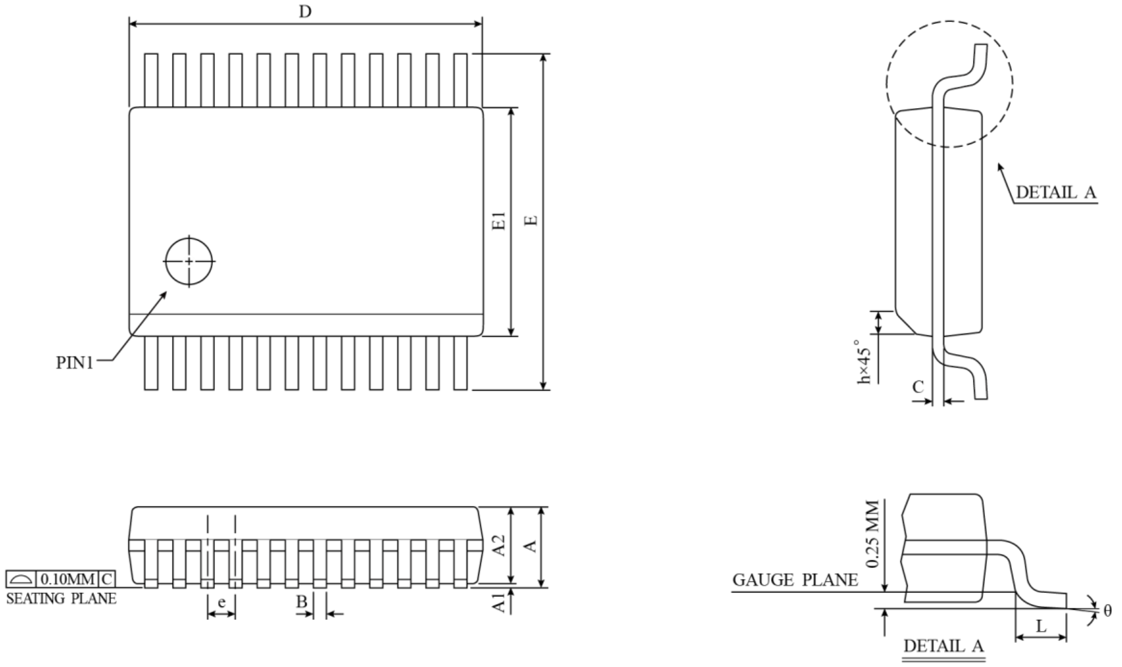


PACKAGING INFORMATION

The ordering information:

Ordering number	Package
TM57PE40-OTP	Wafer / Dice blank chip
TM57PE40-COD	Wafer / Dice with code
TM57PE40-OTP-22	SOP 24-pin (300 mil)
TM57PE40-OTP-32	SSOP 24-pin (209 mil)
TM57PE40-OTP-13	DIP 28-pin (300 mil)
TM57PE40-OTP-23	SOP 28-pin (300 mil)
TM57PE40-OTP-33	SSOP 28-pin (209 mil)
TM57PE40-OTP-24	SOP 32-pin (300 mil)
TM57PE40-OTP-B0	QFN 32-pin (4x4x0.75-0.4mm)

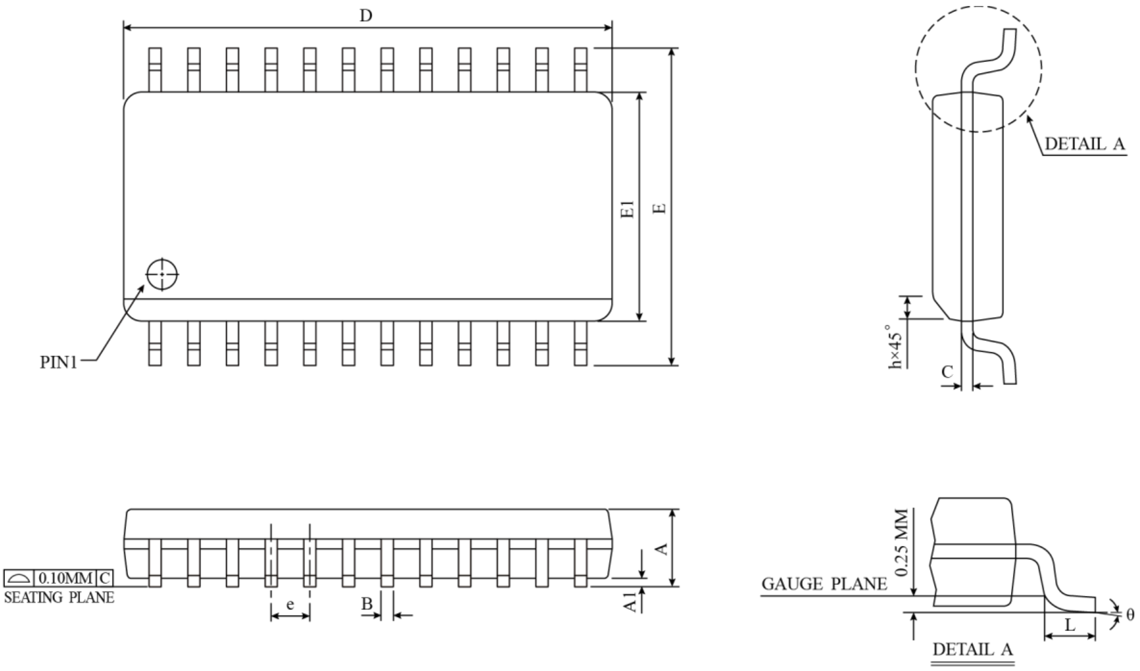
SSOP-24 (209 mil)



SYMBOL	DIMENSION IN MM			DIMENSION IN INCH		
	MIN	NOM	MAX	MIN	NOM	MAX
A	-	-	2.0	-	-	0.079
A1	0.05	-	-	0.002	-	-
A2	1.65	1.75	1.85	0.065	0.069	0.073
B	0.22	0.30	0.33	0.009	0.012	0.013
C	0.09	0.15	0.21	0.004	0.006	0.008
D	7.90	8.20	8.50	0.311	0.323	0.335
E	7.40	7.80	8.20	0.291	0.307	0.323
E1	5.00	5.30	5.60	0.197	0.209	0.220
e	0.65 BSC			0.026 BSC		
L	0.55	0.75	0.95	0.022	0.030	0.038
θ	0°	4°	8°	0°	4°	8°
JEDEC	M0-150 (AG)					

△A * NOTES：DIMENSION 〝D〞 DOES NOT INCLUDE MOLD FLASH
OR PROTRUSIONS, BUT DO INCLUDE MOLD MISMATCH.
MOLD FLASH OR SHALL NOT EXCEED
0.20 MM (0.078 INCH) PER SIDE.

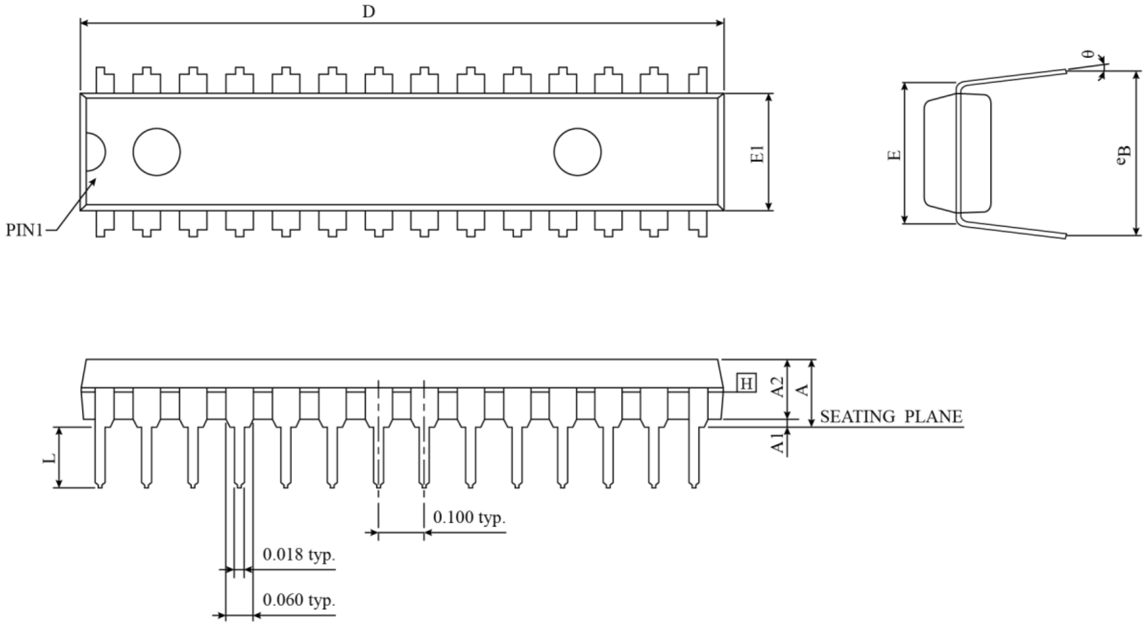
SOP-24 (300 mil)



SYMBOL	DIMENSION IN MM		DIMENSION IN INCH	
	MIN	MAX	MIN	MAX
A	2.35	2.65	0.0926	0.1043
A1	0.10	0.30	0.0040	0.0118
B	0.33	0.51	0.013	0.020
C	0.23	0.32	0.0091	0.0125
D	15.20	15.60	0.5985	0.6141
E	10.00	10.65	0.394	0.491
E1	7.40	7.60	0.2914	0.2992
e	1.27 BSC		0.050 BSC	
h	0.25	0.75	0.010	0.029
L	0.40	1.27	0.016	0.050
θ	0°	8°	0°	8°
JEDEC	MS-013 (AD)			

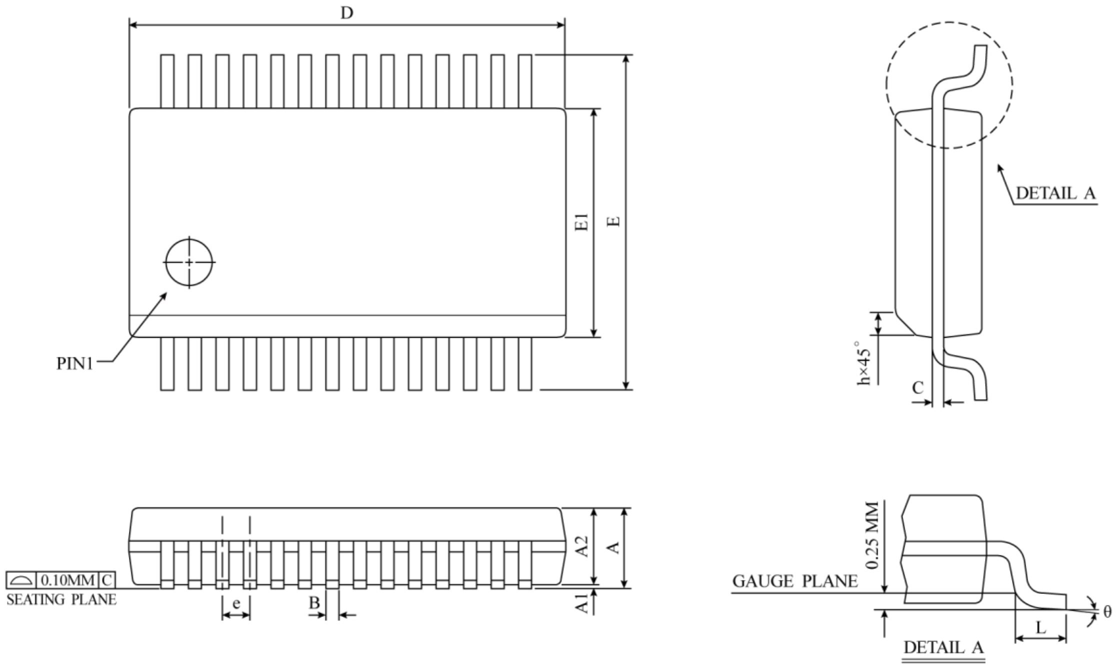
△ * NOTES：DIMENSION 〝D〞 DOES NOT INCLUDE MOLD FLASH, PROTRUSIONS OR GATE BURRS.
MOLD FLASH, PROTRUSIONS AND GATE BURRS SHALL
NOT EXCEED 0.15 MM (0.006 INCH) PER SIDE.

DIP-28 (300 mil)



SYMBOL	DIMENSION IN MM		DIMENSION IN INCH	
	MIN	MAX	MIN	MAX
A	-	4.445	-	0.175
A1	0.381	-	0.015	-
A2	3.175	3.429	0.125	0.135
D	35.179	35.56	1.385	1.400
E	7.874 BSC		0.310 BSC	
E1	7.188	7.442	0.283	0.293
L	3.048	3.556	0.120	0.140
^{e}B	8.382	9.525	0.330	0.375
θ	0°	15°	0°	15°
JEDEC	MS-015 (AH)			

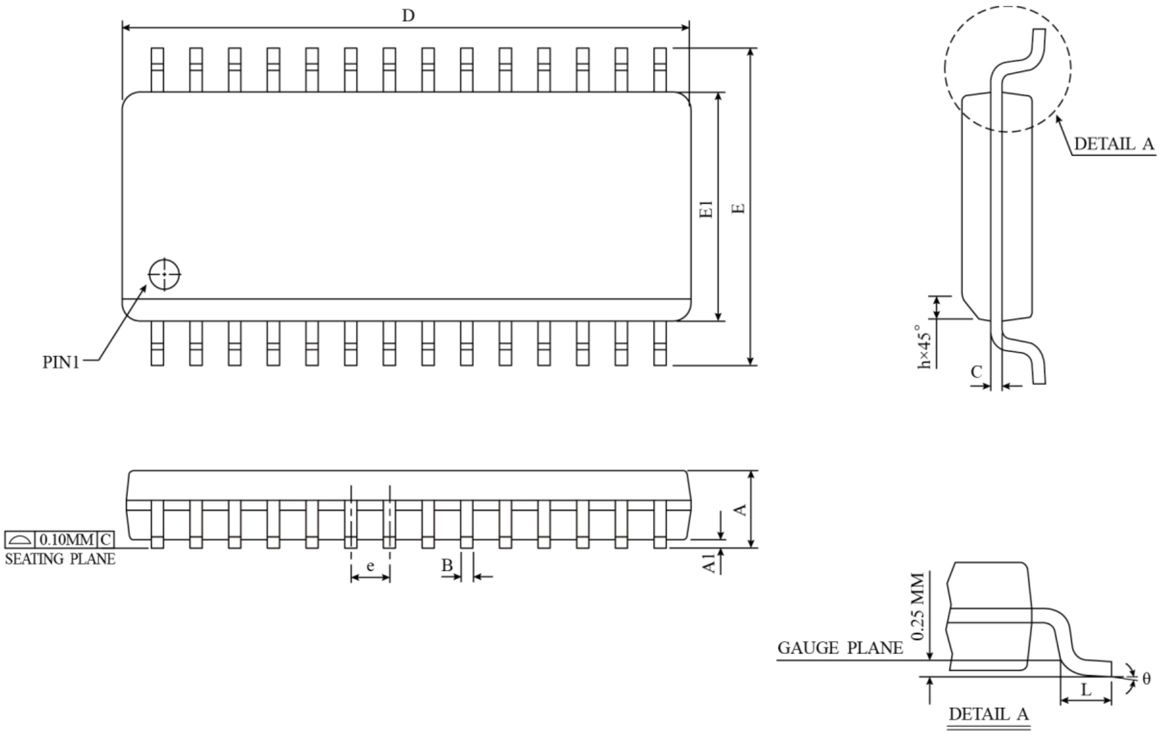
SSOP-28 (209 mil)



SYMBOL	DIMENSION IN MM			DIMENSION IN INCH		
	MIN	NOM	MAX	MIN	NOM	MAX
A	-	-	2.0	-	-	0.079
A1	0.05	-	-	0.002	-	-
A2	1.65	1.75	1.85	0.065	0.069	0.073
B	0.22	0.28	0.33	0.009	0.011	0.013
C	0.09	0.15	0.21	0.004	0.006	0.008
D	9.90	9.98	10.05	0.390	0.402	0.413
E	7.40	7.80	8.20	0.291	0.307	0.323
E1	5.00	5.30	5.60	0.197	0.209	0.220
e	0.65 BSC			0.026 BSC		
L	0.55	0.75	0.95	0.022	0.300	0.038
θ	0°	4°	8°	0°	4°	8°
JEDEC	M0-150 (AH)					

△A * NOTES : DIMENSION " D " DOES NOT INCLUDE MOLD FLASH
OR PROTRUSIONS, BUT DO INCLUDE MOLD MISMATCH.
MOLD FLASH OR SHALL NOT EXCEED
0.20 MM (0.078 INCH) PER SIDE.

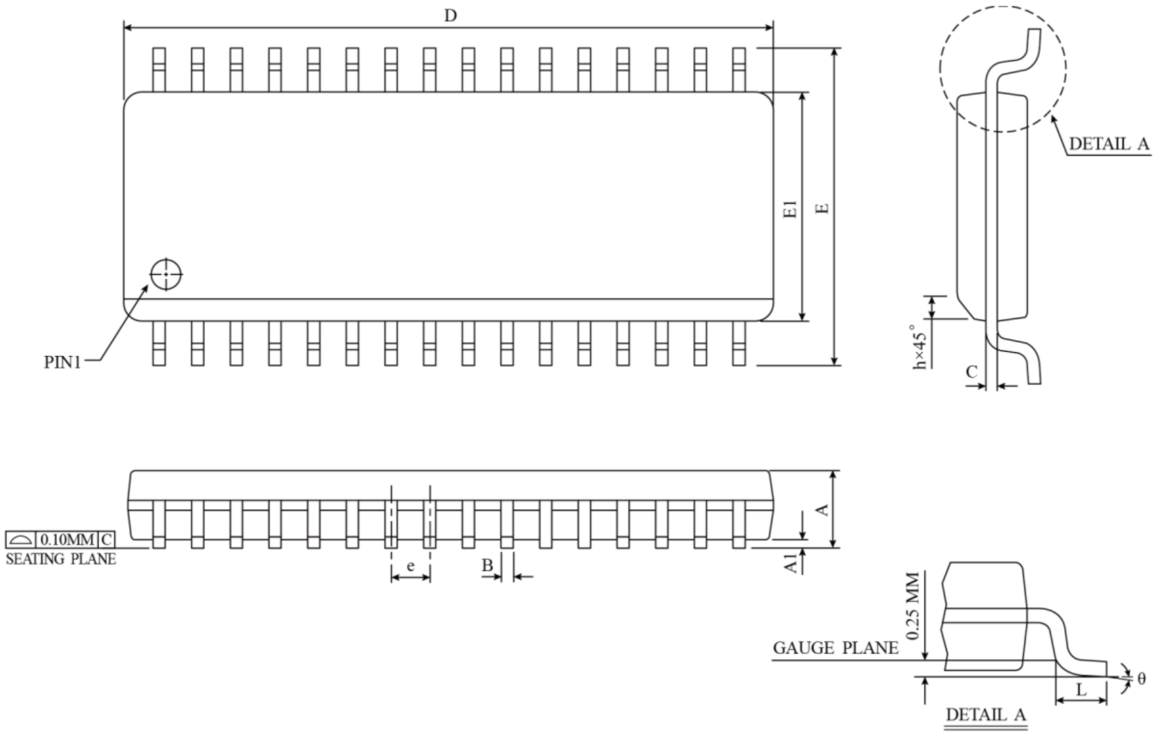
SOP-28 (300 mil)



SYMBOL	DIMENSION IN MM		DIMENSION IN INCH	
	MIN	MAX	MIN	MAX
A	2.35	2.65	0.0926	0.1043
A1	0.10	0.30	0.0040	0.0118
B	0.33	0.51	0.013	0.020
C	0.23	0.32	0.0091	0.0125
D	17.70	18.10	0.6969	0.7125
E	10.00	10.65	0.394	0.491
E1	7.40	7.60	0.2914	0.2992
e	1.27 BSC		0.050 BSC	
h	0.25	0.75	0.010	0.029
L	0.40	1.27	0.016	0.050
θ	0°	8°	0°	8°
JEDEC	MS-013 (AE)			

△ * NOTES : DIMENSION " D " DOES NOT INCLUDE MOLD FLASH, PROTRUSIONS OR GATE BURRS.
MOLD FLASH, PROTRUSIONS AND GATE BURRS SHALL
NOT EXCEED 0.15 MM (0.006 INCH) PER SIDE.

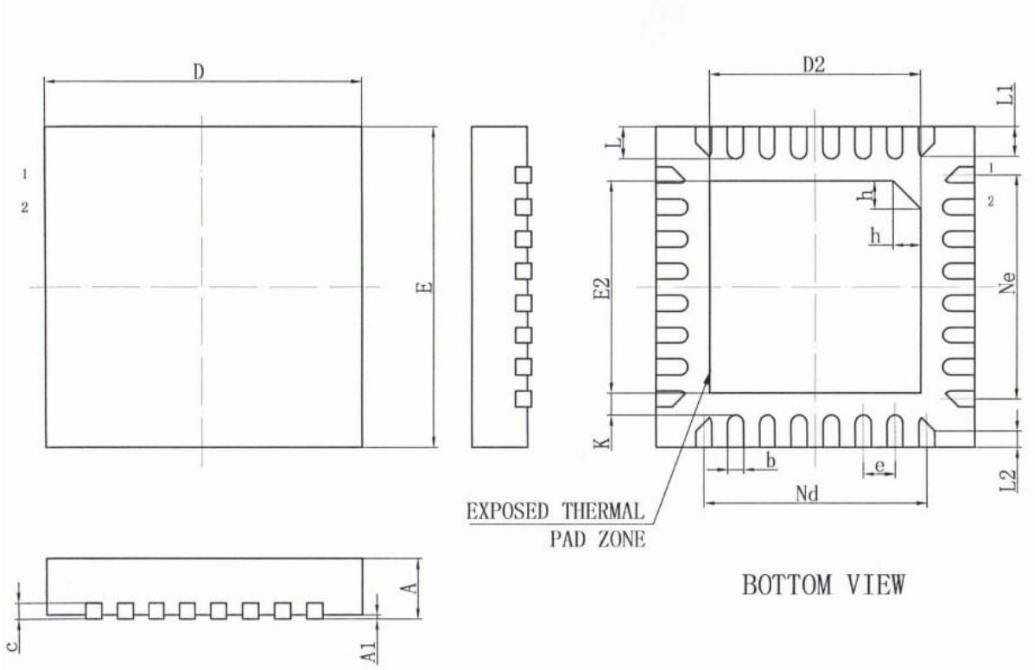
SOP-32 (300 mil)



SYMBOL	DIMENSION IN MM		DIMENSION IN INCH	
	MIN	MAX	MIN	MAX
A	2.35	2.65	0.0926	0.1043
A1	0.10	0.30	0.0040	0.0118
B	0.33	0.51	0.013	0.020
C	0.23	0.32	0.0091	0.0125
D	20.32	20.73	0.800	0.816
E	10.00	10.65	0.394	0.491
E1	7.40	7.60	0.2914	0.2992
e	1.27 BSC		0.050 BSC	
h	0.25	0.75	0.010	0.029
L	0.40	1.27	0.016	0.050
θ	0°	8°	0°	8°

⚠A * NOTES : DIMENSION " D " DOES NOT INCLUDE MOLD FLASH, PROTRUSIONS OR GATE BURRS.
MOLD FLASH, PROTRUSIONS AND GATE BURRS SHALL
NOT EXCEED 0.15 MM (0.006 INCH) PER SIDE.

QFN-32 (4x4x0.75-0.4mm)



SYMBOL	MILLIMETER		
	MIN	NOM	MAX
A	0. 70	0.75	0. 80
A1	0	0. 02	0. 05
b	0. 15	0.20	0. 25
c	0.18	0.20	0.25
D	3. 90	4. 00	4. 10
D2	2. 60	2. 65	2. 70
e	0. 40BSC		
Nd	2. 80BSC		
E	3. 90	4. 00	4. 10
E2	2. 60	2. 65	2. 70
Ne	2. 80BSC		
K	0. 20	-	-
L	0. 35	0. 40	0. 45
L1	0. 30	0. 35	0. 40
L2	0. 15	0. 20	0. 25
h	0. 30	0. 35	0. 40
L/F载体尺寸 (Mil)	112*112		